INFORMATION AND COMPUTATION 108, 51–127 (1994)

Lambda-Calculi for (Strict) Parallel Functions*

GÉRARD BOUDOL

INRIA, BP 93-06902, Sophia-Antipolis Cedex, France

We introduce two λ-calculi and show that they are expressive for two canonical domains of parallel functions. The first calculus is an enrichment of the lazy, call-by-name λ-calculus with call-by-value abstractions and parallel composition, while in the second the usual call-by-name abstractions are disallowed. The corresponding domains are respectively Abramsky's domain $D = (D \to D)_\perp$, a lifted function space, and $D = (D \to_\perp D)_\perp$, a lifted domain of strict functions. These domains are lattices, and we show that the parallelism is adequately represented by the join operator, while call-by-value abstractions correspond to strict functions. The proofs of the results rely on a completeness theorem for the logical presentation of the semantics.

Contents

1. INTRODUCTION

1.1. *The Full Abstraction Problem*

The main theme of this work is the full-abstraction problem for programming languages. This problem, first raised by Milner in [27], can be stated as follows : given a programming language, can we provide a

* This work was partly supported by ESPRIT Basic Research Action 3011 CEDISYS.

0890-5401/94 $6.00

mathematical interpretation of this language such that the resulting semantic equality on programs coincides with operational indistinguishability?

According to Scott [39], what is intended as "mathematical interpretation" is an abstract semantics involving objects that do not refer to a particular way of computing, if not by abstract means, like continuity or stability (cf. Berry *et al.* [7]). Such an abstract semantics would provide a simple intuition about the meaning of programs. But obviously, programs have to be run on a machine, and an abstract semantics has to agree with operational semantics in some way (see again Scott [39] and also [7, 26]). The full abstraction criterion is one way to assess the agreement between a programming language and an abstract semantics for it. Another naturally arising criterion is the *expressivity*, or *completeness*, of the language with respect to an abstract interpretation: a language is expressive (resp. fully expressive, or universal) if all the finite (resp. computable) objects of the domain of interpretation are definable in the language. As a matter of fact, these two criteria are closely related.

We can also turn the full abstraction problem the other way round: given a semantical domain of "objects," can we find a programming language for it, that is, a language in which to express these abstract objects? This is the course we take: we introduce two extensions of the pure, untyped λ-calculus, which are expressive respectively w.r.t. notion of "parallel function" and of "strict parallel function." Our notion of parallel function is quite trivial: we consider domains of continuous functions where any function can be regarded as parallel. We make no attempt at bringing out a notion of "truly parallel"—or conversely of "sequential"—function. For some elements in this direction, see the work of Berry and Curien [7, 16].

Let us recall some well-known results about full abstraction in the case of functional languages (for a survey see Berry *et al.* [7] and Stoughton [44]). This question was investigated by Plotkin [35], who showed that the model of continuous functions over cpo's is not fully abstract for the PCF language, a typed lambda-calculus extended with fixed-points and boolean and arithmetical features. However, this model becomes fully abstract when PCF is enriched with a "parallel or" facility, and the extended language is expressive with respect to the model.

We shall look more closely at the situation of the pure, untyped λ-calculus with respect to the full abstraction problem, concentrating on Abramsky's ideas [3, 4]; for what regards the "classical" λ-calculus we refer to Barendregt's book [5]. Let us first give some general definitions: a programming language consists of a syntax for programs together with an evaluation mechanism, e.g., an abstract machine. These two ingredients are the essence of any operational semantics of programs. The syntactic features of the language provide a notion of *test*: a test is a program

context $C[\ ]$, and two programs are operationally equivalent when they pass exactly the same tests. To observe the success of a test $C[M]$, we have to make it run on the machine, using the evaluation mechanism of the language.

This evaluation mechanism will be formalized by means of a predicate $M \Downarrow K$ meaning "*K is a value of M.*" Usually this also means that K is a normal form of M with respect to some reduction strategy. We shall say that the program M *converges*, in notation $M \Downarrow$, if it has a value, i.e., $\exists K$ $M \Downarrow K$. Conversely divergence is denoted $M \Uparrow$. The notion of convergence, or termination, is, according to Abramsky, the only operationally *observable* property, something like "getting the prompt" on the screen (see also Meyer [26]). Then the succes of a test $C[M]$ is its termination, and operational indistinguishability is given by means of the *testing* or *observation preorder*, whose definition, due to Morris (cf. [5, Exercise 16.5.5]), is as follows:

$$M \sqsubseteq_C N \Leftrightarrow_{\text{def}} \forall C.\, C[M] \Downarrow \Rightarrow C[N] \Downarrow.$$

Clearly, this preorder does not say anything about the abstract nature of the object involved in the langauge. An interpretation, assigning a meaning $[\![M]\!]$ to M in some semantic domain (cpo), is said to be

(i) *adequate* if it induce a precongruence, and assigns no significant value to a divergent program; that is:

$$[\![M]\!] \sqsubseteq [\![N]\!] \Rightarrow \forall C.\, [\![C[M]]\!] \sqsubseteq [\![C[N]]\!] \qquad \text{and} \qquad [\![M]\!] = \bot \Leftrightarrow M \Uparrow;$$

(ii) *fully abstract* if it is adequate and reflects the operational distinctions; that is:

$$M \sqsubseteq_C N \Rightarrow [\![M]\!] \sqsubseteq [\![N]\!].$$

Clearly an adequate semantics does not equate operationally distinct programs; that is, it satisfies

$$[\![M]\!] \sqsubseteq [\![N]\!] \Rightarrow M \sqsubseteq_C N.$$

1.2. *The λ-Calculus: Evaluation Rules and Domain Equations*

Let us now examine the full abstraction problem for the pure λ-calculus. First we should say that it is more appropriate to use this calculus to articulate the workings of an "abstract machine" than of model "programming languages." For this last purpose one should use a typed calculus, with some constants and δ-rules, like PCF. Therefore our discussion concentrates on evaluation rules, and more precisely on parameter passing mechanisms and their semantic counterparts.

For the pure λ-calculus, the adequacy requirement can be stated as follows: recall that this calculus is based upon the assertions

$$\beta\colon (\lambda x M)\, N = M[N/x]$$

$$\nu\colon M = M' \Rightarrow MN = M'N$$

$$\mu\colon N = N' \Rightarrow MN = MN'$$

$$\xi\colon M = M' \Rightarrow \lambda x M = \lambda x M'.$$

Then an adequate semantics of the λ-calculus should satisfy these assertions, and also the requirement concerning divergence. This last property depends on the evaluation mechanism. Let us denote by $\vec{\beta}$, $\vec{\nu}$, $\vec{\mu}$, and $\vec{\xi}$ the conditional rewriting rules obtained by orienting the preceding equations $M = N$ from left to right. Can these rules be considered as providing an evaluation mechanism?

The answer is as follows: first, there is *no* adequate semantics if we allow all these to be evaluation rules. Let us see this quite well-known result in some detail: let $\mathbf{K} = \lambda xy.x$ be the usual "right cancellator," or "left choice," $\Delta = \lambda x(xx)$ the "duplicator," and $\Omega = \Delta\Delta$ the typical divergent term. Since $\mathbf{K}$ is convergent, we should have $[\![\Omega]\!] \neq [\![\mathbf{K}]\!]$ in any adequate semantics. If for any term M we let $F_M = \lambda x((xM)\,\Omega)$, then the term F_M is divergent; therfore we should have $[\![F_M]\!] = [\![F_N]\!]$ for all M and N, but $F_M\mathbf{K} =_\beta M$, hence $[\![M]\!] = [\![N]\!]$, contradicting the existence of an adequate interpretation. The problem is with $\vec{\mu}$, which has to be dropped.

Second, if evaluation is given by $\vec{\beta}$, $\vec{\nu}$, and $\vec{\xi}$, we get a *call-by-name* calculus, since one never evaluates the argument N is an application MN. This is also an *eager*, or *strong* calculus since one may evaluate the body M of a function λxM. In this calculus the (closed) values are the "head normal forms."[1] Note that due to the rule $\vec{\nu}$ we have $M \Uparrow \Rightarrow MN \Uparrow$; therefore $\perp(x) = \perp$ in any adequate semantics. Similarly, due to the rule $\vec{\xi}$ we have $M \Uparrow \Rightarrow \lambda xM \Uparrow$, therefore an adequate semantics should satisfy $\lambda x \perp = \perp$. The strong call-by-name λ-calculus has an adequate, i.e., sensible, and fully abstract semantics, namely Scott's D_∞-interpretation studied by Wadsworth in [45] (cf. [5, Theorem 19.2.9]). However, the D_∞-interpretation is quite strange, since *no* finite non-trivial element of D_∞ is λ-definable; in other words: the "$\beta\nu\xi$-calculus" is not expressive (i.e., it is incomplete) with respect to D_∞. Here the problem is with the $\vec{\xi}$-rule.

We are then left with $\vec{\beta}$ and $\vec{\nu}$, and the situation appears much better; this is Abramsky's *lazy λ-calculus* [3]. This calculus is lazy, or *weak*, since

[1] Warning: this does not mean that the usual notion of "N is a hnf of M" corresponds to $\beta\nu\xi$-evaluation. For instance, if M reduces to N, then $\lambda x(xN)$ is a hnf of $\lambda x(xM)$, but $\lambda x(xM)$ does not $\beta\nu\xi$-reduce to $\lambda x(xN)$. What we get is the "principal" hnf.

one cannot evaluate the body M of an abstraction λxM; it is still a call-by-name calculus since $\vec{\mu}$ is not allowed as an evaluation rule. In the lazy λ-calculus, with "βv-evaluation," there are many more convergent terms than in the "classical" λ-calculus: the (closed) values are "weak head normal forms," that is, abstractions λxM. In other words, the convergent terms are those that can *accept an input*, possibly after some internal evaluation. The typically divergent term is still Ω, but for instance $Y = \lambda x\Omega$, which is clearly "finite," is now strictly greater than Ω in any adequate semantics.

For what regards the lazy λ-calculus, we should first mention that Lévy proposed in [24] an adequate semantics for this calculus, built using syntactic means. Also Longo [25] studied some "very sensible" models of the λ-calculus, dinstinguishing the terms $\lambda x_1 \cdots x_n.\, \Omega$. More roughly, one can dinstinguish in the lazy λ-calculus two kinds of objects: the undefined objects, like Ω, and the truly functional values, like λxM. In other words, any adequate semantics should satisfy $\lambda x \perp \neq \perp$. Accounting for this fact, Abramsky proposed to interpret the lazy λ-calculus in a domain $\boldsymbol{D}_\star$ which is the canonical solution of the equation

$$D = (D \to D)_\perp,$$

where $(X \to Y)$ is the usual domain of continuous functions, extensionally ordered, and $X_\perp$ the lifting construct. So Abramsky's domain consists of all, possibly "non-sequential," continuous functions over itself, plus a non-functional undefined object.

However, the $\boldsymbol{D}_\star$-interpretation of the lazy λ-calculus is not fully abstract—more precisely, it is "over-generous." For instance, it is shown in [4, Corollary 8.1.6] that the step function $\delta : \boldsymbol{D}_\star \to \boldsymbol{D}_\star$ given by

$$\delta(x) = \begin{cases} \mathbf{id} & \text{(the identity function) if } x \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

is *not* definable in the lazy λ-calculus. Abramsky and Ong [4, 32, 33] proposed to enrich the lazy λ-calculus with a combinator $\mathbf{C}$ for *convergence testing*, satisfying

$$\mathbf{C}M \xrightarrow{*} \mathbf{I} \Leftrightarrow M \Downarrow \qquad \text{and} \qquad \mathbf{C}M \Uparrow \Leftrightarrow M \Uparrow,$$

where $\mathbf{I} = \lambda xx$ is the usual identity combinator (thus $[\![\mathbf{C}]\!] = \delta$). However, this seems a little ad hoc, both from a programming language point of view (or more accurately from an abstract machine point of view) and from a denotational point of view. Then the question is: can we find another, more "natural" formulation of the weak λ-calculus with convergence testing? Abramsky remarked that the combinator $\mathbf{C}$ is a variant of the usual "left

cancellator," or "right choice" $\mathbf{F} = \lambda xy.y$; namely, $\mathbf{C}$ is the same as $\mathbf{F}$, but strict in its first argument. Therefore we should extend the λ-calculus so as to be able to define such a strict function which is in fact nearly independent of its argument. Then an appealing solution to the previous question consists in considering not only the usual call-by-name abstraction λxM, but also a *call-by-value abstraction* $\lambda^{\mathrm{v}} xM$, like the one considered by Plotkin in [34]. The corresponding β-rule is

$$\vec{\beta}_{\mathrm{v}}\colon (\lambda^{\mathrm{v}} xM)\, K \to M[K/x] \qquad \text{where } K \text{ is a value.}$$

Obviously we should also add

$$\vec{\mu}_{\mathrm{v}}\colon N \to N' \Rightarrow (\lambda^{\mathrm{v}} xM)\, N \to (\lambda^{\mathrm{v}} xM)\, N'.$$

Clearly we may now define the convergence testing combinator by $\mathbf{C} = \lambda^{\mathrm{v}} x\mathbf{I}$, and conversely the call-by-value abstractions $\lambda^{\mathrm{v}} xM$ could be simulated using the convergence testing combinator, namely by $(\lambda^{\mathrm{v}} M)^{\mathrm{s}} = \lambda x(\mathbf{C}x)\, M^{\mathrm{s}}$ (see also the "call-by-value simulation" of Ong [33]). Then introducing the call-by-value abstractions provides a satisfactory means for testing potential termination. Note that the call-by-name abstractions are still useful, in the untyped calculus, to define non-strict constructs like the "if-then-else" $\mathrm{cond} = \lambda xyz.xyz$. However we shall see that, while call-by-name is not enough to set up an expressive calculus, as we saw above, the standard call-by-value parameter passing mechanism is in some sense sufficiently powerful.

Let us call λ^{nv}-calculus the λ-calculus enriched with call-by-value abstractions, equipped with the evaluation rules $\vec{\beta}$, $\vec{\nu}$, $\vec{\beta}_{\mathrm{v}}$, and $\vec{\mu}_{\mathrm{v}}$, and λ^{v}-calculus the call-by-value λ-calculus, where the ordinary call-by-name abstractions are discarded. It should be clear that, since $N \Uparrow \Rightarrow (\lambda^{\mathrm{v}} xM)\, N \Uparrow$, the call-by-value abstractions must be interpreted as *strict* continuous functions, i.e., such that $f(\bot) = \bot$. In particular, we can adequately interpret in this way the λ^{nv}-calculus in Abramsky's domain $\boldsymbol{D}_{\star}$.

For what regards the λ^{v}-calculus, or *strict λ-calculus*, one can provide an adequate semantics for it by means of *partial functions*, as shown by Plotkin [37] for the typed case, and Moggi [29]. However, it is simpler to use strict functions: let $(X \to_{\bot} Y)$ be the domain of strict continuous functions from X to Y. Obviously one still has to use the lifting construct to reflect the distinction between divergent and convergent λ^{v}-terms, the values being the truly functional terms $\lambda^{\mathrm{v}} xM$. Then the canonical domain for the λ^{v}-calculus is the initial solution $\boldsymbol{D}_{\mathrm{s}}$ of the equation

$$D = (D \to_{\bot} D)_{\bot}.$$

The strict canonical domain $\boldsymbol{D}_s$ has another definition, which is technically more convenient: note that there is a obvious isomorphism $(X_\perp \to_\perp Y) \rightleftharpoons (X \to Y)$, therefore if we denote by V the domain of *values*, that is, $(\boldsymbol{D}_s \to_\perp \boldsymbol{D}_s)$, we may also define our strict domain by the following system of equations:

$$\boldsymbol{D}_s = V_\perp \qquad \text{where}$$
$$V = (V - \boldsymbol{D}_s).$$

Following the terminology of Moggi [30], we could call $\boldsymbol{D}_s$ the domain of *computations*. Note that, since $V = (V \to V_\perp)$, we may also regard the values as *partial functions* over V. In other words, $\boldsymbol{D}_s$ is a partial function space (see Plotkin [37]), enriched with a denotation $\perp$ for the "non-existent" objects. The lifted strict function space $(X \to_\perp Y)_\perp$ is what Abramsky [3, 4] calls the Ladin–Plotkin function space. It is used by Sitaram and Felleisen in [18] to give a fully abstract semantics for a call-by-value PCF enriched with a parallel combinator.

1.3. *Parallel Functions*

The reader should not be surprised to be told that the $\boldsymbol{D}_\star$-interpretation of the λ^{nv}-calculus and the $\boldsymbol{D}_s$-interpretation of the λ^{v}-calculus are not fully abstract: these languages still lack some parallel facility. For instance, it can be seen that the function of $\boldsymbol{D}_\star$ given by

$$\pi(x)(y) = \begin{cases} \mathbf{id} & \text{if } x \neq \perp \text{ or } y \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

is not λ^{nv}-definable. This is clearly a "parallel" function since, although the value of $\pi(x)(y)$ only depends in general on the value of x *or* on the value of y, to compute $\pi(x)(y)$ we have to compute concurrently x and y. The computation of such a function is non-deterministic in that we have to evaluate both arguments and cannot say in advance which will finish first.

Abramsky [3, 4] proposed to enrich the λ-calculus with a parallel convergence testing combinator **P**, whose interpretation is the function π above (see [4, Definition 7.2.7]). This combinator is what Barendregt calls "parallel or;" see [5, Sect. 14.4]. Note that we can define $\mathbf{C} = \lambda x \mathbf{P} x x$. Then the resulting language is expressive with respect to $\boldsymbol{D}_\star$, and the $\boldsymbol{D}_\star$-semantics is fully abstract for it—for all these results, see Abramsky and Ong [4]. This solves the problem stated above: the lazy λ-calculus with parallel convergence testing provides an expressive language for parallel functions. A similar result holds for lazy PCF; see [8, 15].

However, extending the language with this combinator seems again a little ad hoc: instead of adding a special combinator, distinguishing a particular element of the domain, we would like to know which control mechanism could model "parallel computations," and to which property of the domain it would correspond. One may remark that to evaluate $\mathbf{P}MN$ amounts to concurrently evaluating $\mathbf{C}M$ and $\mathbf{C}N$; therefore the parallel convergence testing combinator could be defined as a parallel composition of $\lambda x\lambda y(\mathbf{C}x)$ and $\lambda x\lambda y(\mathbf{C}y)$. This is the solution we adopt, introducing a *parallel composition operator* $(M \parallel N)$. The operational meaning of this operator is as follows: the evaluation of $(M \parallel N)$ initiates two concurrent sub-computations, one for each component, and this term has a value as soon as one of its components has a value. As usual, we formalize a concurrent computation as the non-deterministic interleaving of its sub-computations. Then the evaluation of a parallel combination is given by the following rules:

$$M \to M' \quad \Rightarrow \quad (M \parallel N) \to (M' \parallel N) \text{ and } (N \parallel M) \to (N \parallel M').$$

However, we have to change our mind about the notion of observation, or more precisely about the notion of result, since these rules do not reflect the idea that $(M \parallel N)$ has a value if one of the components, M or N, has a value: a value can no longer be just a normal form, since for instance if M has an infinite computation while N converges, then $(M \parallel N)$ has an infinite computation, hence no normal form, while it should have a value. Therefore we define a value as a term of a special shape, which can be produced at any stage throughout the computation, and not necessarily at the end. More precisely, a value is produced as soon as one concurrent component has terminated. Therefore the syntax of the values is given by the grammar

$$K ::= \lambda xM \mid \lambda^{\vee} xM \mid (K \parallel M) \mid (M \parallel K),$$

where M stands for any term. Intuitively, a value can be regarded as a *set* of functional elements. Then we define the evaluation predicate $M \Downarrow K$, for any closed term M, in such a way that

$$M \Downarrow K \Leftrightarrow K \text{ is a value and } M \xrightarrow{*} K.$$

The notion of convergence, which is the basis of the operational semantics, still means "to have a value," but now this is not the same as "to have a terminating evaluation." However, the convergence property can be adequately interpreted as "to have a terminating sub-computation," and to observe a term means to concurrently observe its components. It is also worth noting that, although our notion of reduction (see Section 2.2)

satisfies the Church–Rosser property, the calculus is non-deterministic: a given term may have several, though compatible, values.

There is a rule missing concerning parallel composition: a term $(M \parallel N)$ is, in general, made of functional units, i.e., abstractions. Then we have to say how it can be applied to an argument. This is the aim of the next rule, which says that the operand is shared between the functional units of the operator:

$$(M \parallel N)\, P \to (MP \parallel NP).$$

Clearly the sharing of data among concurrent programs does not introduce any confict, since the data are distributed to all the components.

Let us see now what is the abstract meaning of concurrency: it can be noted that the canonical domains $D_\star$ and D_s are in fact *algebraic* (*complete*) *lattices*, with all joins.[2] Then it turns out that the parallel composition can be adequately interpreted as the *join* operator, or *sup*, which is typically "non-sequential" or more preisely non-stable; that is,

$$[\![M \parallel N]\!] = [\![M]\!] \sqcup [\![N]\!].$$

Then the parallel composition operator is also called the *concurrent join*, and the calculi we propose for parallel functions are respectively the λ_j^{nv}-calculus, that is, the *call-by-name* and *call-by-value* λ-*calculus with join*, and λ_j^{v}-calculus, the *call-by-value* or *strict* λ-*calculus with join*.

Using the idea that a parallel function is the join of some more basic functional elements, it is easy for instance to "program" a parallel disjunction. Let us see this in more detail: first recall that **K** and **F** may be regarded as the *truth values*, respectively true and false. Then parallel disjunction **O** is just the join $(\mathbf{V}_l \parallel \mathbf{V}_r)$ of "left sequential disjunction" $\mathbf{V}_l = \lambda xy.(x\mathbf{K})\, y$, that is, $\mathbf{V}_l = \lambda xy.\mathrm{cond}\, x\mathbf{K}y$, and "right sequential disjunction" $\mathbf{V}_r = \lambda xy.(y\mathbf{K})\, x$, or $\mathbf{V}_r = \lambda xy.\mathrm{cond}\, y\mathbf{K}x$ (cf. Curien [16]). One should note that evaluating a parallel combination always yields another parallel combination; therefore, though $(\mathbf{K} \parallel \Omega) = \mathbf{K}$ is a valid semantic equation, we cannot get simply the value **K** in evaluating $\mathbf{O}MN$, for instance.

Let us now return to the full abstraction problem: recall that this property relates a semantical preorder with a testing preorder, and that a test is just a program context. For instance, in the λ_j^{nv}-calculus, the test $(\lambda x[\,\cdot\,])\,\Omega$ can be used to distinguish the term **I**, which passes the test successfully, from $\mathbf{C}x$, which does not (recall that $\mathbf{C} = \lambda^{v} x\mathbf{I}$). In fact, the same test can be used to distinguish $(\lambda^{v} xM)\, x$ from M, for any closed convergent term M. Then equality of call-by-value terms in the λ_j^{nv}-calculus is different from equality Plotkin's λ_v-calculus [34], where, for M closed,

[2] As we shall see, the canonical domains are also (lower) powerdomains, with the join as the union operator.

$(\lambda^{\mathrm{v}} xM)\,x = M$ by β-conversion since in this calculus a variable is a value. However, using $\lambda_{\mathrm{j}}^{\mathrm{v}}$-tests, one cannot detect any difference between these two terms—it would be interesting to show that our λ^{v}-calculus and Plotkin's λ_{v}-calculus are the same, from the testing point of view. One can give a semantical account of this $\lambda_{\mathrm{j}}^{\mathrm{v}}$-operational semantics, by considering only environments $\rho : \boldsymbol{X} \to \boldsymbol{D}_{\mathrm{s}}$ such that $\rho(x) \neq \bot$ for any variable x.

However, to develop in a similar manner the theory of the two calculi, we use an unrestricted notion of environment in both cases. Therefore the terms $(\lambda^{\mathrm{v}} xM)\,x$ and M, where M is a closed convergent $\lambda_{\mathrm{j}}^{\mathrm{v}}$-term, are semantically different. This difference must be tested operationally; to this end we keep on with a call-by-name feature in the $\lambda_{\mathrm{j}}^{\mathrm{v}}$-calculus, namely substitution. More precisely, we use tests of the form $C[M\sigma]$, where σ is a substitution. Note that there is in the $\lambda_{\mathrm{j}}^{\mathrm{v}}$-calculus an asymmetry between the two ways of closing a term, namely abstraction and substitution: in $\lambda^{\mathrm{v}} xM$ the bound variable x only stands for "existent" objects, i.e., convergent terms, while one can substitute any term for a free variable. As a matter of fact, distinguishing between the ranges of free and bound variables is a standard matter in existence logic; see, for instance, Scott [41].

Since we assume that the test are provided by the syntax of the language, we should in principle include substitution as a fully explicit computational mechanism, as is done by Abadi *et al.* in [1], for instance. This means that we should introduce terms of the form $M\sigma$, that is, general closures, where σ is a syntactic "environment," and extend the abstract machine to deal with them—we could also use the notation (let $x = N$ in M) for a term which is evaluated as $M[N/x]$ (like $(\lambda xM)\,N$ in the $\lambda_{\mathrm{j}}^{\mathrm{nv}}$-calculus). However, we content ourselves here with "semi-explicit" substitutions. More precisely, substitutions will be used two ways: as we said, we consider test of the form $C[M\sigma]$, where $M\sigma$ is not a term, but denotes the result of applying the substitution σ to the term M—this is supposed to be provided by an outer evaluation procedure. Then, to avoid any problem with "fresh names" and α-conversion, we replace the ordinary abstractions λxM, and also the strict ones, by *closures*, i.e., pairs $\langle \lambda xM, \sigma \rangle$ made of a functional value and an environment. Then the calculi $\lambda_{\mathrm{j}}^{\mathrm{nv}}$ and $\lambda_{\mathrm{j}}^{\mathrm{v}}$ for parallel functions that will be studied here have the following syntax: the grammar of the first is

$$M ::= x \mid \langle \lambda xM, \sigma \rangle \mid \langle \lambda^{\mathrm{v}} xM, \sigma \rangle \mid (MM) \mid (M \parallel M)$$

$$\sigma ::= \varepsilon \mid [M/x]\,\sigma,$$

while we disallow the use of $\langle \lambda xM, \sigma \rangle$ for the strict sub-calculus $\lambda_{\mathrm{j}}^{\mathrm{v}}$. The usual abstractions λxM and $\lambda^{\mathrm{v}} xM$ are notations for closures where the environment part is the empty substitution ε.

Since one may define the parallel convergence combinator by $\mathbf{P} = (\lambda x\lambda y(\mathbf{C}x) \parallel \lambda x\lambda y(\mathbf{C}y))$, where $\mathbf{C} = \lambda^{\nu} x\mathbf{I}$, it is not surprising, in view of Abramsky's results, that the following holds:

THEOREM 1. *The $\lambda_j^{n\nu}$-calculus is complete with respect to the canonical domain $\boldsymbol{D}_\star$. Moreover, the $\boldsymbol{D}_\star$-interpretation of this calculus is fully abstract.*

We show that a similar result holds in the strict case:

THEOREM 2. *The λ_j^{ν}-calculus is complete with respect to the canonical domain $\boldsymbol{D}_s$. Moreover, the $\boldsymbol{D}_s$-interpretation of this calculus is fully abstract.*

Before giving an idea of the proofs, let us say a few words about domains. Scott originally proposed to work with lattices, and we saw that this seems quite convenient for dealing with "parallel" functions. However the meaning of the $\top$ element in Scott's work was not entirely clear. Sometimes, as in [39], $\top$ is an "over-determined," inconsistent element; sometimes it is the most multivalued element, as in [40]. I think both interpretations are right: as the join of all values, $\top$ is clearly the most multivalued element. But then $\top$ does not provide much information: anything is true of it. This is reflected in the syntax: in the $\lambda_j^{n\nu}$-calculus we can define an "ogre," namely

$$Y_\star =_{\text{def}} (\lambda x\lambda y(xx))(\lambda x\lambda y(xx)).$$

This combinator is the "best" one, greater than any other. But it is not a very interesting one: since $Y_\star \to \lambda y Y_\star$, the ogre can repeatedly accept any input value you propose, but then it simply ignores it and returns the prompt.

The use of lattices in semantics was criticized, for instance by Plotkin in [36], for the reason that some natural identities, concerning the "if-then-else" construct, fail if the $\top$ element is admitted. The first expected identity, namely

$$\text{cond } XYY = Y$$

(where X and Y are values), refers implicitly to a notion of "type": we cannot guarantee anything if the first argument of the if-then-else is not a "boolean." Indeed we have cond $Y_\star XX = Y_\star$, but also cond$(\mathbf{K} \parallel \mathbf{F})\, XY = (X \parallel Y)$, hence cond$(\mathbf{K} \parallel \mathbf{F})\, XX = X$. The other expected identity, that is,

$$\text{cond } X(\text{cond } XYZ)\, V = \text{cond } XYV,$$

assumes implicitly that X is either true or false (or undefined). Obviously this equation does not hold in a non-deterministic language, where X may be the join of the truth values.

Another criticism against lattices, or more precisely against the flat lattice model of PCF, is presented by Bloom [9] and Meyer [26]. Meyer argued that, although the mathematical results they provide are quite nice, these lattices should be rejected because any expressive language for them has to be "unreasonable"—this meaning that the evaluation mechanism has to be inherently non-deterministic (non-confuent). As we indicated above, our notion of reduction is indeed confluent, but the notion of observation is non-deterministic, since we cannot predict the relative speed of execution of the concurrent components. Unlike Bloom [9], we do not regard "single-valuedness as a essential criterion." One may also wonder whether our results could be adapted for PCF, using a powerdomain (see Section 3.1.1) for the domain of integers, instead of the flat lattice used in [35, 9].

1.4. *The Proofs of Full Abstraction*

As we indicated above, the main point of this paper is to show that the interpretations of the λ_j^{nv} and λ_j^{v} calculi, respectively in the domains $\boldsymbol{D}_\star$ and $\boldsymbol{D}_s$, provide adequate and fully abstract semantics for these languages. For the first one, this is more or less an adaptation of Abramsky and Ong's results [4]. As far as we know, the second full abstraction result is new. We prove both these results here, since the proofs follow exactly the same lines. Now let us see how these proofs are articulated:

(1) The first step is to give a concrete presentation of the canonical domains. This is done using Scott's information systems [43]. Then it can be observed, following the work of Coppo *et al.* [12, 13], that another equivalent concrete presentation is by means of *filters of logical formulae.* This formalizes Scott's idea that "elements of a domain can be identified with the sets of propositions true of them," which was already realized in Coppo and co-workers [11, 6]. This is also a simple instance of Abramsky's theory of "domains in logical form" [2]. Then the domains are *logical domains.* The logic for $\boldsymbol{D}_\star$ is

$$\phi ::= \omega \mid (\phi \to \phi) \mid (\phi \wedge \phi),$$

together with an axiomatization of the entailment relation $\phi \leqslant \psi$ meaning intuitively "ϕ implies ψ." This is the logic of Abramsky [3], and also, apart from the propositional variables, of Dezani-Ciancaglini and

Margaria [17]. For the strict domain, the formulae belong to a subset of the previous logic, given by

$$\psi ::= \omega \mid \zeta \mid (\psi \wedge \psi)$$
$$\zeta ::= (\omega \to \omega) \mid (\zeta \wedge \zeta) \mid (\zeta \to \psi),$$

with a slightly different axiomatization of the entailment relation. In this logic one is not allowed to form the proposition $\omega \to \psi$ unless ψ is the trivial formula ω, corresponding to the fact that functions are strict.

Since the abstract meaning $[\![M]\!]$ of a term M is now a set of formulae, we can recast the interpretation of the calculi using the logical presentation of the domains, and we get "typing systems," that is, logical natural deduction system allowing one to prove sequents of the form

$$x_1 : \phi_1, \ldots, x_n : \phi_n \vdash M : \phi.$$

It turns out that, for closed terms,

$$\vdash M : \phi \Leftrightarrow \phi \in [\![M]\!].$$

Therefore if we define an "assertional preorder," or syntactic logical preorder, by

$$M \sqsubseteq_{\mathscr{S}} N \Leftrightarrow_{\text{def}} \forall H \forall \phi. \vdash M : \phi \Rightarrow \vdash N : \phi,$$

we have

$$[\![M]\!] \subseteq [\![N]\!] \Leftrightarrow M \sqsubseteq_{\mathscr{S}} N.$$

Then we prove some properties of the logical systems, which are used to show that the semantical preorders are precongruences preserved by instantiation; that is,

$$M \sqsubseteq_{\mathscr{S}} N \Rightarrow \begin{cases} \forall C. C[M] \sqsubseteq_{\mathscr{S}} C[N] & \text{and} \\ \forall \sigma. M\sigma \sqsubseteq_{\mathscr{S}} N\sigma. \end{cases}$$

In particular we have to show that the usual *cut* rule, that is,

$$\frac{H \vdash N : \psi,\ x : \psi,\ H \vdash M : \phi}{H \vdash M[N/x]\,\varepsilon : \phi},$$

is valid in our logical systems—note that this is the rule for typing (let $x = N$ in M). In Coppo's typing systems a kind of converse property, which we call *paste*, also holds. This property says that a proof of $H \vdash M\sigma : \phi$ is always composed of proofs of intermediary assertions for the terms

assigned to the variables in the substitution σ, and a proof that M satisfies ϕ under the corresponding hypothesis:

$$H \vdash M[N_1/x_1] \cdots [N_k/x_k]\, \varepsilon : \phi$$
$$\Rightarrow \exists \phi_1, \ldots, \phi_k \begin{cases} H \vdash N_1 : \phi_1, \ldots, H \vdash N_k : \phi_k & \text{and} \\ x_1 : \phi_1, \ldots, x_k : \phi_k \vdash M : \phi. \end{cases}$$

The cut and paste properties are used to show that the semantical preorder is preserved by instantiation. The paste property also allows one to prove that "reduction is decreasing;" that is, for closed terms,

$$M \Downarrow K \quad \text{and} \quad \vdash K : \phi \Rightarrow \vdash M : \phi.$$

It is easy to see that a (closed) value K satisfies the convergence formula; that is, $\vdash K : \omega \to \omega$. Then this establishes a first half of the *computational adequacy* property for the abstract interpretations, namely

$$M \Downarrow \Rightarrow [\![M]\!] \neq \bot.$$

(2) To prove the converse we use a "realizability interpretation" of the formulae as sets of closed terms, namely

$$\models M : \omega \Leftrightarrow_{\text{def}} \text{true}$$
$$\models M : (\phi \wedge \psi) \Leftrightarrow_{\text{def}} \models M : \phi \text{ and } \models M : \psi$$
$$\models M : (\psi - \phi) \Leftrightarrow_{\text{def}} M \Downarrow \text{ and } \forall R.\ \models R : \psi \Rightarrow \models (MR) : \phi.$$

It is worth noting the similarity between this interpretation and Hindley's "F-semantics" for type schemes [19, 17]. Note that a term realizes $\psi \to \phi$ when it has a truly functional character, i.e., it converges, and when applied to arguments realizing ψ then any result it returns realizes ϕ.

This interpretation is extended to arbitrary terms by defining $H \models M : \phi$, which roughly means $\models M\sigma : \phi$ for any substitution σ satisfying the hypothesis H (and such that $M\sigma$ is closed). Then we prove the *soundness* of the logical systems with respect to the realizability interpretation; that is,

$$H \vdash M : \phi \Rightarrow H \models M : \phi.$$

Since we clearly have $\models M : \omega \to \omega \Leftrightarrow M \Downarrow$, we get the second half of the computational adequacy property, namely

$$[\![M]\!] \neq \bot \Leftrightarrow M \Downarrow.$$

Together with the previous results, this shows that our abstract interpretations are *adequate*; therefore the semantic preorders are stronger than the testing preorders:

$$M \sqsubseteq_{\mathscr{S}} N \Rightarrow M \sqsubseteq_{\mathscr{C}} N.$$

(3) To prove the converse implication we define another logical preorder associated with the realizability interpretation,

$$M \sqsubseteq_{\mathscr{L}} N \Leftrightarrow_{\text{def}} \forall H\ \forall \phi.H \models M : \phi \Rightarrow H \models N : \phi$$

and it is quite easy to show (see [3, 4]) that this is implied by the testing preorder:

$$M \sqsubseteq_{\mathscr{C}} N \Rightarrow M \sqsubseteq_{\mathscr{L}} N.$$

Then the full abstraction theorem is a consequence of the implication

$$M \sqsubseteq_{\mathscr{L}} N \Rightarrow M \sqsubseteq_{\mathscr{S}} N.$$

This in turn results from the *completeness* of the logical systems with respect to the realizability interpretation; that is,

$$H \models M : \phi \Rightarrow H \vdash M : \phi.$$

The completeness theorem is proved using Hindley's method [19, 20], by means of a refined deduction theorem. The key fact for completeness is that any compact element of the domain $\boldsymbol{D}_{\star}$ (resp. $\boldsymbol{D}_{\mathrm{s}}$) is $\lambda_{\mathrm{j}}^{\mathrm{nv}}$-definable (resp. $\lambda_{\mathrm{j}}^{\mathrm{v}}$-definable)—in other words, our languages are expressive with respect to their abstract interpretations, namely,

$$\forall \phi\ \exists \mathbf{M}_{\phi}.\ \vdash \mathbf{M}_{\phi} : \psi \Leftrightarrow \phi \leqslant \psi.$$

This is the only result for which we need the concurrent join $\|$, together with the call-by-value abstractions (as a test for existence, required in the realizability interpretation of $\psi \to \phi$). The essential role of this definability result is not a surprise: although it was not formulated in logical terms in the pioneering work of Plotkin [35] and Milner [28], the completeness of the language was recognized as a crucial property for the full abstraction problem. The idea of the "characteristic terms" $\mathbf{M}_{\phi}$ is due to Abramsky [3] (see also [14], where Coppo remarked that the proof of completeness relies upon the existence of "characteristic values" for each type). However, we should point out that using the concurrent join considerably simplifies their definition (see [4, Theorem 7.1.3]), since to represent a conjunction we can let $\mathbf{M}_{\phi \wedge \psi} = (\mathbf{M}_{\phi} \| \mathbf{M}_{\psi})$.

Note. Although the results are simple adaptations of more or less standard ones, most of the proofs are given in full detail. The proof that the testing preorder is stronger than the logical preorder, that is,

$$M \sqsubseteq_{\mathscr{C}} N \Rightarrow M \sqsubseteq_{\mathscr{L}} N,$$

is directly adapted from [3, 4]. For the equivalence of the "assertional" and logical preorders, that is,

$$M \sqsubseteq_{\mathscr{S}} N \Leftrightarrow M \sqsubseteq_{\mathscr{L}} N,$$

we use proof techniques mainly due to Coppo and co-workers [6, 10, 12], Hindley [19, 20], and Krivine [21]. Finally, we give only hints for the proof of the equivalence of the "assertional" and semantical preorders, that is,

$$[\![M]\!] \sqsubseteq [\![N]\!] \Leftrightarrow M \sqsubseteq_{\mathscr{S}} N.$$

We refer the interested reader to [13] for more details, but we hope that he/she will tink it is an instructive exercise to find out the arguments. We mainly take into consideration the "constructive" presentation of the canonical interpretations: in fact these interpretations are given by means of the logical natural deduction systems.

2. The λ_j^{nv} and λ_j^{v}-Calculi

2.1. *Syntax and Evaluation*

To build the terms of our calculi, we assume given an infinite set X of variables. As just said in the introduction, the λ_j^{nv}-calculus is given by the grammar

$$(\Lambda_j^{nv}) \qquad M ::= x \mid \langle \lambda x M, \sigma \rangle \mid \langle \lambda^{v} x M, \sigma \rangle \mid (MM) \mid (M \parallel M)$$

$$(\Sigma) \qquad \sigma ::= \varepsilon \mid [M/x]\,\sigma,$$

where x is any variable.

We call $\parallel$ the *concurrent join*, or parallel composition, $\langle \lambda x M, \sigma \rangle$ a *call-by-name closure*, $\langle \lambda^{v} x M, \sigma \rangle$ *a call-by-value closure*, and ε the empty (or identity) substitution. We use $M, N, P, Q, R, \ldots$ to denote terms of Λ_j^{nv}. The set of call-by-value terms, or λ_j^{v}-terms, built without using the closures $\langle \lambda x M, \sigma \rangle$, is Λ_j^{v}.

Notational Convention. In what follows most of the concepts we introduce have two versions, one for the λ_j^{nv}-calculus and one for the

$\lambda_j^{\vee}$-calculus. We use the symbol "$*$" to refer to the first version as in $\boldsymbol{D}_*$, while "s" refers to the second version, as in $\boldsymbol{D}_s$.

We often use the standard abstraction construct λxM as an abbreviation for $\langle \lambda xM, \varepsilon \rangle$, and similarly for $\lambda^{\vee} xM$. As usual, we omit some parentheses, denoting (MN) by MN, and more generally $(\cdots(MN_1)\cdots N_k)$ by $MN_1 \cdots N_k$. We use the standard abbreviation for sequences of λ-abstractions, namely $\lambda x_1 \cdots x_n . M$ for $\lambda x_1 \cdots \lambda x_n M$.

We use a *structural ordering* on $\Lambda_j^{n\vee} \cup \Sigma$, denoted $s \propto t$, meaning that s is a syntactic component of t. The formal definition of this ordering should be obvious. In particular we have $\sigma \propto \langle \lambda xM, \sigma \rangle$ and $\sigma \propto \langle \lambda^{\vee} xM, \sigma \rangle$, and $\sigma \propto [M/x]\sigma$ and $M \propto [M/x]\sigma$. It should be clear that this ordering is noetherian; that is, there is no strictly decreasing infinite chain for it. This allows us to use structural induction, that is, induction on the structural ordering, in definitions and proofs. The following combinators were already introduced, at least in their non-strict version:

identity:	$\mathbf{I} = \lambda xx$	and	$\mathbf{I}_s = \lambda^{\vee} xx$
duplicator:	$\Delta = \lambda x(xx)$	and	$\Delta_s = \lambda^{\vee} x(xx)$
divergence:	$\Omega = \Delta\Delta$	and	$\Omega_s = \Delta_s \Delta_s$
least value:	$\Upsilon = \lambda x\Omega$	and	$\Upsilon_s = \lambda^{\vee} x\Omega_s$
truth values:	$\mathbf{T} = \lambda x\lambda yx = \mathbf{K}$	and	$\mathbf{F} = \lambda y\lambda xx$
	$\mathbf{T}_s = \lambda^{\vee} x\lambda^{\vee} yx = \mathbf{K}_s$	and	$\mathbf{F}_s = \lambda^{\vee} y\lambda^{\vee} xx$
ogre:	$\Upsilon_* = (\lambda x\lambda y(xx))(\lambda x\lambda y(xx))$.		

The substitution $[M/x]\sigma$ can also be regarded as an *environment*, providing the value M for x, and acting as a stack: if σ also assigns a value for x, this value is ignored. This is formalized by defining the value $\sigma(x)$ of the variable x in the environment σ, as follows:

$$\varepsilon(x) = x$$

$$([M/y]\sigma)(x) = \begin{cases} M & \text{if } y = x \\ \sigma(x) & \text{otherwise.} \end{cases}$$

The set $\mathrm{fv}(M)$ of *free variables* of M is given by

$$\mathrm{fv}(x) = \{x\}$$

$$\mathrm{fv}(\langle \lambda xM, \sigma \rangle) = \{v \mid \exists z . z \in \mathrm{fv}(M) - \{x\} \text{ and } v \in \mathrm{fv}(\sigma(z))\} = \mathrm{fv}(\langle \lambda^{\vee} xM, \sigma \rangle)$$

$$\mathrm{fv}(MN) = \mathrm{fv}(M) \cup \mathrm{fv}(N) = \mathrm{fv}(M \parallel N).$$

As usual, a term M is said to be *closed* if $\mathrm{fv}(M) = \varnothing$. The set of closed terms is denoted $(\Lambda_j^{nv})^\diamond$. Similarly $(\Lambda_j^{v})^\diamond$ is the set of closed λ_j^v-terms.

We shall regard as *given*, by an outer evaluation procedure, the operations of *composition* of substitutions $(\sigma \circ \rho)$ and *application* $M\sigma$ of substitution σ to a term M. These operations are supposed to satisfy

$$x\sigma = \sigma(x)$$

$$\langle \lambda x M, \sigma \rangle \rho = \langle \lambda x M, (\sigma \circ \rho) \rangle$$

$$\langle \lambda^v x M, \sigma \rangle \rho = \langle \lambda^v x M, (\sigma \circ \rho) \rangle$$

$$(MN)\sigma = (M\sigma N\sigma)$$

$$(M \parallel N)\sigma = (M\sigma \parallel N\sigma)$$

$$(\varepsilon \circ \rho) = \rho$$

$$([M/x]\sigma) \circ \rho = [M\rho/x](\sigma \circ \rho).$$

One may note that the evaluation mechanism for $M\sigma$ does not have to be especially sophisticated. In particular, there is no need for converting bound variables using new names. It is a very easy exercise to show the following:

LEMMA 2.1. (i) $M\varepsilon = M$ *and* $\sigma \circ \varepsilon = \sigma$.

(ii) $(M\sigma)\rho = M(\sigma \circ \rho)$ *and* $(\rho \circ \sigma_0) \circ \sigma_1 = \rho \circ (\sigma_0 \circ \sigma_1)$.

We use the standard notion of *context*: a λ_j^{nv}-context, or simply context (resp. λ_j^v-context, or strict context) is a term built using the λ_j^{nv}-constructs (resp. the λ_j^v-constructs), and possibly a new constant $\square$, the *hole*. However, we do not allow the hole occur in the environment part of a closure context. Then the syntax of λ_j^{nv}-contexts is given by the grammar

$$C ::= \square \mid x \mid \langle \lambda x C, \sigma \rangle \mid \langle \lambda^v x C, \sigma \rangle \mid (CC) \mid (C \parallel C).$$

The λ_j^v-contexts are given by a similar grammar, where the call-by-name closure are discarded. Filling the hole with the context B in the context C gives the context $C[B]$ (the definition, by structural induction, is obvious). Note in this operation, some free variables of B may be bound by the context C, e.g., $(\lambda x \square)[x] = \lambda x x$. We say that a context C *closes* the term M if $C[M]$ is closed.

Let us now define the *evaluation mechanism* for λ_j^{nv}-terms, formalized as a relation $M \Downarrow K$ meaning "*K is a partial normal form of M*", or a *value* of M. In principle we should define two such relations $\Downarrow_\star$ and $\Downarrow_v$, but in fact the latter is just restriction of the former to the set of λ_j^v-terms. The partial normal forms, or values, ranged over by $K, J, L, \ldots$, are closures

$\langle \lambda xM, \sigma \rangle$ or $\langle \lambda^{v} xM, \sigma \rangle$, and parallel combinations including at least one partial normal form. Then the syntax of partial normal forms is given by the grammar

$$K ::= \langle \lambda xM, \sigma \rangle \mid \langle \lambda^{v} xM, \sigma \rangle \mid (K \parallel M) \mid (M \parallel K),$$

where M stands for any terms.

We only define evaluation for *closed terms*, i.e., $M \in (\Lambda_{j}^{nv})^{\diamond}$, by means of the rules given below. Our first two rules for evaluation just assert that the closures are their own normal form:

$$\text{R.1.1:} \quad \langle \lambda xM, \sigma \rangle \Downarrow \langle \lambda xM, \sigma \rangle \qquad \text{R.1.2:} \quad \langle \lambda^{v} xM, \sigma \rangle \Downarrow \langle \lambda^{v} xM, \sigma \rangle.$$

The next rules describe how to evaluate an application MN: to do so we have first to evaluate the operator M into a partial normal form, if any. Then there are three cases: if this partial normal form for M is a call-by-name closure $\langle \lambda xR, \rho \rangle$, then we have to evaluate the body R of this closure in the updated environment $[N/x]\,\rho$; that is,

$$\text{R.2.1:} \quad \frac{M \Downarrow \langle \lambda xR, \rho \rangle,\ R[N/x]\,\rho \Downarrow K}{(MN) \Downarrow K}.$$

If the partial normal form for the operator M is a call-by-value closure $\langle \lambda^{v} xR, \rho \rangle$, then we have to evaluate the operand N into a partial normal form J, and then to evaluate the body R in the updated environment $[J/x]\,\rho$; that is,

$$\text{R.2.2:} \quad \frac{M \Downarrow \langle \lambda^{v} xR, \rho \rangle,\ N \Downarrow J,\ R[J/x]\,\rho \Downarrow K}{(MN) \Downarrow K}.$$

Note that these two rules implicitly call for the substitution procedure that has to return a term for $R[N/x]\,\rho$ or $R[J/x]\,\rho$. One should remark that the rule R.2.2 is not exactly the one of Plotkin's λ_{v}-calculus [34], since here a variable is not taken as a "value."

The last case of the evaluation of an application MN is the one where the partial normal form we get for the operator M is a parallel combination $(M_0 \parallel M_1)$. In this case the evaluation proceeds by distributing the operand N to the concurrent components; that is,

$$\text{R.2.3:} \quad \frac{M \Downarrow (M_0 \parallel M_1),\ (M_0 N \parallel M_1 N) \Downarrow K}{(MN) \Downarrow K}.$$

Let us see an example of evaluation using these rules: since $(\lambda y(xx))[M/x]\,\varepsilon = \langle \lambda y(xx), [M/x]\,\varepsilon \rangle$ and $\lambda x \lambda y(xx) \Downarrow \lambda x \lambda y(xx)$, one has

$$Y_{\star} \Downarrow \langle \lambda y(xx), [\lambda x \lambda y(xx)/x]\,\varepsilon \rangle.$$

Moreover

$$(xx)[X/y][\lambda x\lambda y(xx)/x]\,\varepsilon = Y_\star;$$

therefore

$$Y_\star X \Downarrow \langle \lambda y(xx),\ [\lambda x\lambda y(xx)/x]\,\varepsilon \rangle.$$

Our last rules for evaluation concern the concurrent join. They simply assert that to evaluate $(M \parallel N)$, one has to evaluate M or N and join the partial normal forms; that is,

$$\text{R.3.1:}\ \frac{M \Downarrow K}{(M \parallel N) \Downarrow (K \parallel N)} \qquad \text{R.3.2:}\ \frac{N \Downarrow J}{(M \parallel N) \Downarrow (K \parallel J)}.$$

$$\text{R.3.3:}\ \frac{M \Downarrow K,\ N \Downarrow J}{(M \parallel N) \Downarrow (K \parallel J)}.$$

It should be clear that although we have some choice in evaluating $(M \parallel N)$, the rules R3 would preserve a Church–Rosser property; that is,

$$M \Downarrow K_0 \quad \text{and} \quad M \Downarrow K_1 \Rightarrow \exists K.\, K_0 \Downarrow K \quad \text{and} \quad K_1 \Downarrow K.$$

However, this property fails to hold, owing to the rule R.2.2: if M has two distinct partial normal forms K_0 and K_1, as it is generally the case if $M = (M_0 \parallel M_1)$, then $(\lambda^{\vee} x\lambda y R)\, M$ converges to the partial normal forms $\langle \lambda y R,\ [K_0/x]\,\varepsilon \rangle$ and $\langle \lambda y R,\ [K_1/x]\,\varepsilon \rangle$, and these two have no common partial normal form. However, we shall see that the non-determinism of the evaluation mechanism does not create real conflicts: in fact, evaluating the environment part of a closure would not be incorrect. This is shown in the next section. To this end we shall use the following facts about evaluation:

LEMMA 2.2. (i) *if K is a partial normal form then $K \Downarrow K$;*

(ii) *$M \Downarrow K$ and $K \Downarrow L \Rightarrow M \Downarrow L$;*

(iii) *$(MN) \Downarrow K \Leftrightarrow \exists L.\, M \Downarrow L$ and $(LN) \Downarrow K$.*

Proof. The first point should be obvious. One shows the second one by induction on the inference of $M \Downarrow K$. Let us see the case of R.3.1, that is, $M = (P \parallel Q)$ and $K = (J \parallel Q)$ with $P \Downarrow J$. Then $K \Downarrow L$ can only be proved using R.3.1, R.3.2, or R.3.3; in this last case we have $L = (J' \parallel Q')$ with $J \Downarrow J'$ and $Q \Downarrow Q'$, therefore $P \Downarrow J'$ by induction hypothesis, and $M \Downarrow L$ by R.3.3. The other cases are left to the reader.

For what regards the third point, the implication $(MN) \Downarrow K \Rightarrow \exists L.\, M \Downarrow L$ and $(LN) \Downarrow K$ should be clear, since $(MN) \Downarrow K$ can only be proved using R.2.1, R.2.2, or R.2.3, and each of these rules implies $M \Downarrow L$

for some L (note that, by the first point of the lemma, $L \Downarrow L$). Conversely one proves $M \Downarrow L$ and $(LN) \Downarrow K \Rightarrow (MN) \Downarrow K$ by induction on the proof of $M \Downarrow L$. The details are omitted. ∎

We say that a closed term M is *convergent* (resp. *divergent*), in notation $M \Downarrow$ (resp. $M \Uparrow$), if it has a value, that is, there exists K such that $M \Downarrow K$ (resp. there is no such K). It should be clear that $\Omega \Uparrow$ for instance, since a proof of $\Omega \Downarrow K$ would call for a shorter proof of the same fact. It is easy to see that the following holds:

$$M \Uparrow \Rightarrow MN \Uparrow.$$

Moreover in the call-by-value calculus λ_j^v we also have

$$N \Uparrow_s \Rightarrow MN \Uparrow_s.$$

2.2. *Reduction. Preliminary Results*

In this section we establish some results regarding substitutions and a notion of reduction. Let us first define a congruence relation $M \equiv N$, meaning "M and N are the same, up to some garbage collection." The "garbage collection" operates on the environment part of the closures. Typically we have $\langle \lambda x M, \sigma \rangle \equiv \langle \lambda x M, \rho \rangle$ if $\sigma(z) = \rho(z)$ for all $z \in \mathrm{fv}(M) - \{x\}$, and similarly for the strict closures. Then this relation, called gc-equivalence, is the least one such that

(i) $M \equiv M$ and

$M \equiv M' \Rightarrow M' \equiv M$ and

$M \equiv M''$ and $M'' \equiv M' \Rightarrow M \equiv M'$

(ii) $M \equiv M' \Rightarrow \begin{cases} (MN) \equiv (M'N) \text{ and } (NM) \equiv (NM') \\ (M \parallel N) \equiv (M' \parallel N) \text{ and } (N \parallel M) \equiv (N \parallel M') \end{cases}$

(iii) $(\forall z \in \mathrm{fv}(M) - \{x\}.\, \sigma(z) \equiv \rho(z)) \Rightarrow \begin{cases} \langle \lambda x M, \sigma \rangle \equiv \langle \lambda x M, \rho \rangle \\ \langle \lambda^v x M, \sigma \rangle \equiv \langle \lambda^v x M, \rho \rangle. \end{cases}$

An obvious remark is that $M \equiv N \Rightarrow \mathrm{fv}(M) = \mathrm{fv}(N)$.

LEMMA 2.3. (i) $\forall x \in \mathrm{fv}(M).\, \sigma(x) \equiv \rho(x) \Rightarrow M\sigma \equiv M\rho$.

(ii) $M \equiv N \Rightarrow M\sigma \equiv N\sigma$.

Proof. For the first point one proceeds by induction on M. The only case deserving some consideration is the case of closures. So let us assume that $M = \langle \lambda x N, \varsigma \rangle$, for instance. Then $M\sigma = \langle \lambda x N, \varsigma \circ \rho \rangle$, and for all

$z \in \mathrm{fv}(N) - \{x\}$ we have $(\varsigma(z))\,\sigma \equiv (\varsigma(z))\,\rho$ by the induction hypothesis, $(\varsigma(z))\,\sigma = (\varsigma \circ \sigma)(z)$ by Lemma 2.1, and similarly $(\varsigma(z))\,\rho = (\varsigma \circ \rho)(z)$; therefore $M\sigma \equiv M\rho$ by definition of the equivalence.

For the second point, one proceeds by induction on the definition of $M \equiv N$. If this is obtained using the last clause of the definition, we have $M = \langle \lambda xR, \varsigma \rangle$ and $N = \langle \lambda xR, \rho \rangle$ with $\varsigma(z) \equiv \rho(z)$ for all $z \in \mathrm{fv}(R) - \{x\}$ (or M and N are strict closures). Since by Lemma 2.1 we have $(\varsigma \circ \sigma)(z) = (\varsigma(z))\,\sigma$ and $(\rho \circ \sigma)(z) = (\rho(z))\,\sigma$, by the induction hypothesis $(\varsigma \circ \sigma)(z) \equiv (\rho \circ \sigma)(z)$, therefore $M\sigma \equiv N\sigma$ by definition of the equivalence. All the other cases are easy. ∎

An obvious consequence is

COROLLARY 2.4. *$M \equiv N$ and $\forall x \in \mathrm{fv}(M).\,\sigma(x) \equiv \rho(x) \Rightarrow M\sigma \equiv N\rho$.*

For instance, we have

$$x \notin \mathrm{fv}(M) \Rightarrow M[R/x]\,\sigma \equiv M\sigma.$$

Note also that if M is closed then $M\sigma \equiv M$, since $M\sigma \equiv M\varepsilon = M$.

Now we introduce a notion of *reduction* $M \rhd N$, between closed terms, to be read "N is a reduction of M." This relation, extending the usual β-reduction—except that the ξ-rule is limited to reduction in the environment part of the closures—provides us with an alternative presentation of the operational semantics. Since we want to show a Church–Rosser property for reduction, we permit simultaneous contraction of several redexes in a given term, as in Tait and Martin-Löf's notion of parallel reduction. Then the reduction relation is the least relation on closed terms satisfying

(i) $M \rhd M$

(ii) $(\forall z \in \mathrm{fv}(M) - \{x\}.\,\sigma(z) \rhd \rho(z)) \Rightarrow \begin{cases} \langle \lambda xM, \sigma \rangle \rhd \langle \lambda xM, \rho \rangle \\ \langle \lambda^{\vee} xM, \sigma \rangle \rhd \langle \lambda^{\vee} xM, \rho \rangle \end{cases}$

(iii) $M \rhd M'$ and $N \rhd N' \Rightarrow \begin{cases} (MN) \rhd (M'N') \\ (M \parallel N) \rhd (M' \parallel N') \end{cases}$

(iv) $\langle \lambda xM, \sigma \rangle\, N \rhd M[N/x]\,\sigma$ and

$\langle \lambda^{\vee} xM, \sigma \rangle\, K \rhd M[K/x]\,\sigma$ if K is a partial normal form, and

$(M_0 \parallel M_1)\, N \rhd (M_0 N \parallel M_1 N)$.

An obvious remark is:

Remark 2.5. If K is a partial normal form and $K \rhd N$ then N is a partial normal form.

Another observation is that gc-equivalent terms may be reduced to each other:

LEMMA 2.6. *For M closed $M \equiv N \Rightarrow M \rhd N$ and $N \rhd M$.*

Proof. By induction on the definition of $M \equiv N$, trivial. ∎

To establish the Church–Rosser property, we need first to show the following:

LEMMA 2.7. *If $\sigma_0(x) \rhd \sigma_1(x)$ for any $x \in \mathrm{fv}(M)$ then $M\sigma_0 \rhd M\sigma_1$.*

Proof. By induction on the term M. Let us just see the case $M = \langle \lambda xR, \rho \rangle$. Then, for $i = 0, 1$, $M\sigma_i = \langle \lambda xR, \rho_i \rangle$ where $\rho_i = (\rho \circ \sigma_i)$, that is, $\rho_i(z) = (\rho(z))\,\sigma_i$ (see Lemma 2.1). By the induction hypothesis $\rho_0(z) \rhd \rho_1(z)$ for any $z \in \mathrm{fv}(R) - \{x\}$, hence $M\sigma_0 \rhd M\sigma_1$ by definition of the reduction relation. All the other cases are left to the reader. ∎

Now we show that the reduction relation has the Church–Rosser property:

LEMMA 2.8 (The Church–Rosser Property). *$M \rhd K_0$ and $M \rhd K_1 \Rightarrow \exists K.\, K_0 \rhd K$ and $K_1 \rhd K$.*

Proof. This is trivial if $K_0 = M$ or $K_1 = M$ or $K_0 = K_1$. Thus we assume $K_i \neq M$ and $K_0 \neq K_1$. We proceed by induction on M, examining the possible critical pairs. In fact we can, by symmetry, restrict our investigations to the following cases:

- If $M = \langle \lambda xR, \sigma \rangle$ then $K_i = \langle \lambda xR, \sigma_i \rangle$ with $\sigma(z) \rhd \sigma_i(z)$ for any $z \in \mathrm{fv}(R) - \{x\}$. By the induction hypothesis, for any $z \in \mathrm{fv}(R) - \{x\}$ there exists $\rho(z)$ such that $\sigma_i(z) \rhd \rho(z)$. Therefore we may let $K = \langle \lambda xR, \rho \rangle$. The case of $M = \langle \lambda^{\vee} xR, \sigma \rangle$ is entirely similar.

- If $M = (PQ)$ with $K_0 = (P'Q')$, $P \rhd P'$, and $Q \rhd Q'$, then there are four cases. The first one, where $K_1 = (P''Q'')$, $P \rhd P''$, and $Q \rhd Q''$, is obvious: one simply uses the induction hypothesis. If $P = \langle \lambda xR, \sigma \rangle$ and $K_1 = R[Q/x]\,\sigma$, then $P' = \langle \lambda xR, \rho \rangle$ with $\sigma(z) \rhd \rho(z)$ for any $z \in \mathrm{fv}(R) - \{x\}$. Let $K = R[Q'/x]\,\rho$; then $K_0 \rhd K$ by definition of the reduction relation, and $K_1 \rhd K$ by the previous lemma. The case of $P = \langle \lambda^{\vee} xR, \sigma \rangle$ and $K = R[Q/x]\,\sigma$, where Q is a partial normal form, is entirely similar (from Remark 2.5 Q' is also a partial normal form). Finally, if $P = (P_0 \parallel P_1)$ and $K_1 = (P_0 Q \parallel P_1 Q)$, then it is easy to see that $P' = (P'_0 \parallel P'_1)$ with $P_i \rhd P'_i$. We let $K = (P'_0 Q' \parallel P'_1 Q')$; clearly $K_0 \rhd K$ since $K_0 = (P'_0 \parallel P'_1)\,Q'$. Moreover, $P_i Q \rhd P'_i Q'$ by definition of the reduction relation; therefore $K_1 \rhd K$.

• If $M = (P \parallel Q)$ then clearly $K_i = (P_i \parallel Q_i)$ with $P \rhd P_i$ and $Q \rhd Q_i$, and we simply use the induction hypothesis. ▮

Let us denote by $N \lhd M$ the converse of the reduction relation. Now we introduce a relation of *conversion*, $M \bowtie N$, between closed terms. As usual this is just the reflexive and transitive closure of $\rhd \cup \lhd$; that is,

$$M \bowtie N \Leftrightarrow_{\text{def}} M(\rhd \cup \lhd)^* N.$$

Clearly this convertibility relation is an equivalence, and an obvious consequence of the Church–Rosser property is

COROLLARY 2.9. *$M \bowtie N \Leftrightarrow \exists R.\, M \rhd^* R$ and $N \rhd^* R$.*

To conclude this section we relate the reduction relation and the evaluation relation introduced in the previous section. More precisely, we show that if a term has a value, then it reduces to this value, and that reduction and evaluation commute:

LEMMA 2.10. (i) *$M \Downarrow K \Rightarrow M \rhd^* K$;*

(ii) *$M \rhd N$ and $N \Downarrow K \Rightarrow \exists L.\, M \Downarrow L$ and $L \rhd K$.*

Proof. We prove the first point by induction on the inference of $M \Downarrow K$. Let us just see the case of R.2.2: $M = (PQ)$, $P \Downarrow \langle \lambda^{\vee} xR, \sigma \rangle$, $Q \Downarrow J$, and $R[J/x]\,\sigma \Downarrow K$. Then by the induction hypothesis $P \rhd^* \langle \lambda^{\vee} xR, \sigma \rangle$ and $Q \rhd^* J$, therefore $M \rhd^* \langle \lambda^{\vee} xR, \sigma \rangle\, J$, hence $M \rhd^* R[J/x]\,\sigma$, since J is a partial normal form, and finally $M \rhd^* K$ by the induction hypothesis on $R[J/x]\,\sigma \Downarrow K$. The other cases are left to the reader.

For the second point we proceed by induction on the proof of $N \Downarrow K$, and then by cases on the definition of $M \rhd N$. In fact we can factorize three cases for $M \rhd N$. First, the result is trivial if $M = N$. Moreover, if $M = \langle \lambda xP, \rho \rangle\, Q$ and $N = P[Q/x]\,\rho$, then $M \Downarrow K$ by R.2.1, and similarly if $M = \langle \lambda^{\vee} xP, \rho \rangle\, Q$ and $N = P[Q/x]\,\rho$, where Q is a partial normal form, then $M \Downarrow K$ by R.2.2 since $Q \Downarrow Q$. These case are not taken into account in the following proof.

• If $N \Downarrow K$ is proved using R.1.1, then $N = K = \langle \lambda xR, \sigma \rangle$. The only possibility for inferring $M \rhd \langle \lambda xR, \sigma \rangle$—apart from the cases mentioned above—is $M = \langle \lambda xR, \rho \rangle$ with $\rho(z) \rhd \sigma(z)$ for any $z \in \text{fv}(R) - \{x\}$. This case is trivial since $M \Downarrow M$.

• The proof is similar if $N \Downarrow K$ is inferred using R.1.2.

• If $N \Downarrow K$ is proved using R.2.2 (we do not investigate the case of R.2.1, which is entirely analogous) then $N = (PQ)$ with $P \Downarrow \langle \lambda^{\vee} xR, \sigma \rangle$,

$Q \Downarrow J$, and $R[J/x]\,\sigma \Downarrow K$. Again the only possibility for inferring $M \rhd N$ we have to consider is $M = (P'Q')$ with $P' \rhd P$ and $Q' \rhd Q$. Then by the induction hypothesis there exist P'' and J' such that $P' \Downarrow P'' \rhd \langle \lambda^{\vee} xR, \sigma \rangle$ and $Q' \Downarrow J' \rhd J$. Since P'' is a partial normal form we have $P'' = \langle \lambda^{\vee} xR, \rho \rangle$ with $\rho(z) \rhd \sigma(z)$ for any $z \in \mathrm{fv}(R) - \{x\}$. Then by Lemma 2.7 $R[J'/x]\,\rho \rhd R[J/x]\,\sigma$; therefore by the induction hypothesis there exists L such that $R[J'/x]\,\rho \Downarrow L \rhd K$, and $M \Downarrow L$ by R.2.2.

• If $N \Downarrow K$ is proved by means of R.2.3 then $N = (PQ)$ with $P \Downarrow (P_0 \parallel P_1)$ and $(P_0 Q \parallel P_1 Q) \Downarrow K$. Assume that $M = (P'Q')$ with $P' \rhd P$ and $Q' \rhd Q$. By the induction hypothesis there exists P'' such that $P' \Downarrow P'' \rhd (P_0 \parallel P_1)$. Since P'' is a partial normal form we have $P'' = (P_0' \parallel P_1')$ with $P_i' \rhd P_i$; therefore $P_i' Q \rhd P_i Q'$, hence $(P_0' Q \parallel P_1' Q) \rhd (P_0 Q' \parallel P_1 Q')$. By the induction hypothesis there exists L such that $(P_0' Q \parallel P_1' Q) \Downarrow L$ and $L \rhd K$; therefore $M \Downarrow L$ by R.2.3.

The remaining cases (R.3.1, R.3.2, and R.3.3) are left to the reader. ∎

COROLLARY 2.11. $M \bowtie N \Rightarrow M \Downarrow \Leftrightarrow N \Downarrow$.

Proof. If $M \Downarrow K$ then $M \rhd^* K$ by the previous lemma, therefore $M \bowtie N$ and $M \Downarrow K \Rightarrow N \bowtie K$. By Corollary 2.9 there exists R such that $N \rhd^* R$ and $K \rhd^* R$. Since K is a partial normal form, R is also a partial normal form, hence $R \Downarrow R$. An obvious consequence of the previous lemma is $N \rhd^* R$ and $R \Downarrow \Rightarrow N \Downarrow$. ∎

The results about conversion we use later can be summarized as follows:

PROPOSITION 2.12. *The conversion relation* $\bowtie$ *has the following properties*:

(i) $M \equiv N \Rightarrow M \bowtie N$;

(ii) $M \bowtie N \Rightarrow M \Downarrow \Leftrightarrow N \Downarrow$;

(iii) $M \bowtie N \Rightarrow (MR) \bowtie (NR)$;

(iv) $\langle \lambda xM, \sigma \rangle\, N \bowtie M[N/x]\,\sigma$

(v) $N \Downarrow \Rightarrow \langle \lambda^{\vee} xM, \sigma \rangle\, N \bowtie [N/x]\,\sigma$.

Proof. The first and second points have already been established. The third and fourth points result directly from the definition of the reduction relation. The last point is true since $N \Downarrow$ implies $N \rhd^* J$ for some partial normal form J; therefore we have $\langle \lambda^{\vee} xM, \sigma \rangle\, N \rhd^* \langle \lambda^{\vee} xM, \sigma \rangle\, J \rhd M[J/x]\,\sigma$, and $M[N/x]\,\sigma \rhd^* M[J/x]\,\sigma$ (Lemma 2.7). ∎

3. Semantics

3.1. *The Canonical Domains*

In this section we introduce the domains $\boldsymbol{D}_\star$ and $\boldsymbol{D}_s$ that we use to interpret the λ_j^{nv} and λ_j^{v} calculi. These domains are respectively the initial solutions of the equations

$$D = (D \to D)_\perp$$

and

$$D = V_\perp \qquad \text{where}$$
$$V = (V \to D),$$

where $X_\perp$ is the usual *lifting* of X, adjoining a new bottom element to X, and $(X \to Y)$ is the domain of *continuous functions* from X to Y. It is worth noting that the initial solutions of these equations yield *prime algebraic lattices* (cf. [31]).

This section is quite allusive—in particular no proof is given. The reason is that we regard the *logical interpretation*, given below, as the actual interpretation. Here we begin with some abstract concepts, and introduce the logical content of these concepts by gradual changes in the notations: first we recall some definitions concerning lattices, which are dealt with using Scott's information systems [43]. This provides us with an explicit presentation of the canonical domains $\boldsymbol{D}_\star$ and $\boldsymbol{D}_s$, using standard means to solve recursive domain equations (see for instance Larsen and Winskel [22]). Finally we get a logical characterization à la Coppo [12, 13] of our domains by giving a syntax for the canonical information systems.

3.1.1. *Algebraic Lattices*

We first recall some standard definitions concerning lattices. A *complete lattice* is a partial order $(L, \sqsubseteq)$ such that each subset X of L has a least upper bound $\bigsqcup X$. The least upper bound of two elements x and y is their *join* $x \sqcup y$. A complete lattice has a least and a greatest element, namely $\perp = \bigsqcup \varnothing$ and $\top = \bigsqcup L$. The *lifting* construct builds a complete lattice $L_\perp$ out of a given complete lattice L: this simply consists in adding a new bottom element. A subset X of L is *directed* if for any finite subset Y of X there exists $x \in X$ such that $\bigsqcup Y \sqsubseteq x$ (therefore a directed set is non-empty). Given two complete lattices L_0 and L_1, a function $f: L_0 \to L_1$ is *continuous* iff it preserves the joins of directed subsets, that is iff for any directed subset X of L we have $f(\bigsqcup X) = \bigsqcup f(X)$. The set $(L_0 \to L_1)$ of continuous functions, extensionally ordered, i.e., $f \sqsubseteq g \Leftrightarrow_{\text{def}} \forall x f(x) \sqsubseteq_1 g(x)$, is a complete lattice.

An element c of a complete lattice L is *compact* iff for any subset X of L such that $c \sqsubseteq \bigsqcup X$ there exists a finite subset Y of X such that $c \sqsubseteq \bigsqcup Y$. We use $\mathscr{K}(L)$ to denote the set of compact elements of L. A complete lattice is *algebraic* if any element e of L is the join of the compact points it dominates:

$$e = \bigsqcup \{c \mid c \in \mathscr{K}(L) \text{ and } c \sqsubseteq e\}.$$

The representation of algebraic lattices by means of information systems rests upon the following well-known characterization: every algebraic lattice is isomorphic to the family of ideals of a join-semilattice with bottom element. Let us spell out this statement: a join semilattice with bottom is a partial order such that any finite subset has a join. An ideal is a directed *cone*, that is, a directed subset X such that $x \in X$ and $y \sqsubseteq x \Rightarrow y \in X$. By a *family* we mean any set of sets, ordered by inclusion.

A fundamental fact about algebraic lattices is that a continuous function f from an algebraic lattice L_0 to an algebraic lattice L_1 is uniquely determined by the set of pairs

$$\text{graph}(f) = \{(c, d) \mid c \in \mathscr{K}(L_0), d \in \mathscr{K}(L_1), \text{ and } d \sqsubseteq_1 f(c)\}.$$

We could say that a pair $(c, d) \in \text{graph}(f)$ is an "element of information" about the function f, requiring the information c on the input to produce the information d on the output. Note that the information (c, d) is better than (c', d') if it requires less—i.e., $c \sqsubseteq_0 c'$—to produce more—i.e., $d' \sqsubseteq_1 d$. This is the basic idea of the concrete representation of continuous functions over algebraic lattices that is used below.

As we shall see, our canonical domains have a property stronger than algebraicity; namely, they are prime algebraic lattices. Let us recall the definition, taken from Nielsen, *et al.* [31]. First we say that an element p of a lattice L is *prime* iff $p \sqsubseteq x \sqcup y$ implies $p \sqsubseteq x$ or $p \sqsubseteq y$. Note that $\perp$ is prime according to our terminology. We denote by $\mathscr{P}(L)$ the set of prime elements of L, and by $\mathscr{K}\mathscr{P}(L)$ the set of compact prime elements—these are, apart from $\perp$, the "complete primes" of [31]. Then a complete lattice is *prime algebraic* if any element e of L is the join of the compact primes it dominates:

$$e = \bigsqcup \{c \mid c \in \mathscr{K}\mathscr{P}(L) \text{ and } c \sqsubseteq e\}.$$

Note that if c is compact then it is the join of a *finite* number of compact primes: that is, $c = p_1 \sqcup \cdots \sqcup p_n$ for some $p_i \in \mathscr{K}\mathscr{P}(L)$.

We do not use here the characterization of prime algebraic lattices given by Nielsen *et al.*, but it is worth noting that any prime algebraic lattice is

a kind of powerdomain. Given any poset $(P, \sqsubseteq)$, we could define the *lower powerdomain*, or Hoare powerdomain $P^\flat$ of this poset as the family of ideals of the pre-ordered set $(\mathbf{Fin}(P), \sqsubseteq^\flat)$, where $\mathbf{Fin}(P)$ is the set of finite non-empty subsets of P, and $u \sqsubseteq^\flat v$ is given by

$$u \sqsubseteq^\flat v \Leftrightarrow_{\text{def}} \forall x \in u\, \exists y \in v.x \sqsubseteq y.$$

For instance, if the poset $(P, \sqsubseteq)$ is discrete, that is, $\sqsubseteq$ is the equality, then its lower powerdomain is just (isomorphic to) the powerset of P, that is, the set of non empty subsets of P, ordered by inclusion. The *union* in $P^\flat$ is given by

$$X \cup Y =_{\text{def}} \{w \mid \exists u \in X\, \exists v \in Y\, w \sqsubseteq^\flat u \cup v\}.$$

Usually one uses this definition of powerdomain when P is the set of compact elements of some domain. Here we use it as follows: given a prime algebraic lattice $(L, \sqsubseteq)$ we define its *prime powerdomain* $\mathscr{H}(L)$ as the lower powerdomain of the poset $(\mathscr{KP}(L), \sqsubseteq)$ of compact prime elements of L. Then we have:

LEMMA. *Any prime algebraic lattice is isomorphic to its own prime powerdomain*: $L = \mathscr{H}(L)$. *Moreover, the union in* $\mathscr{H}(L)$ *corresponds to the join in* L.

3.1.2. *Lattice Information Systems*

We can give a more concrete formulation of the aforementioned characterization of algebraic lattices, using Scott's notion of information system. Since we are interested in lattices, the information systems we use do not involve the consistency predicate. Moreover, we disallow the empty set from entering into consideration. Then our definition is as follows:

DEFINITION 3.1 (Lattice Information Systems). A lattice information system is a structure $S = (A, \Delta, \vdash)$ where A is the set of atoms, $\Delta \in A$, and $\vdash$ is a relation over finite non-empty subsets of A satisfying:

D1: $u \vdash \{\Delta\}$

D2: $u \vdash v$ and $v \vdash w \Rightarrow u \vdash w$

D3: $u \subseteq v \Rightarrow v \vdash u$

D4: $u \vdash v$ and $u \vdash w \Rightarrow u \vdash v \cup w$.

Since we only use lattice information systems from now on we omit the "lattice" qualification. Let us introduce some notations: the elements of $\mathbf{Fin}(A)$, that is, the *finite non-empty* subsets of A, are ranged over by $u, v, w, \ldots$. Sometimes we denote $u \vdash v$ by $a_1, \ldots, a_n \vdash b_1, \ldots, b_k$ when $u = \{a_1, \ldots, a_n\}$ and $v = \{b_1, \ldots, b_k\}$. For instance $a_1, \ldots, a_n \vdash \Delta$ in any information system.

A basic example is the following: let $(K, \sqsubseteq)$ be a join semilattice with bottom $\perp$. Then it is easy to see that the structure $(K, \perp, \vdash)$, where $\vdash$ is given by

$$a_1, \ldots, a_n \vdash b_1, \ldots, b_k \Leftrightarrow_{\text{def}} (b_1 \sqcup \cdots \sqcup b_k) \sqsubseteq (a_1 \sqcup \cdots \sqcup a_n),$$

is an information structure.

According to Scott, we may interpret the atoms $a \in A$ of an information structure as the "propositions" that can be made about the "elements" described by the information structure. The "proposition" Δ means true, and the *deduction* (or entailment) relation $u \vdash v$ formalizes the idea that if the "propositions" of u are true of some element, then the "propositions" of v are also true of the same element. In fact, an element is just the set of propositions true of it. Then the domain of S is the *family* $(\mathscr{D}(S), \subseteq)$ of *non-empty, deductively closed* subsets of A; that is,

$$X \in \mathscr{D}(S) \Leftrightarrow_{\text{def}} \begin{cases} X \subseteq A \text{ and } \Delta \in X & \text{and} \\ u \subseteq X \text{ and } u \vdash v \Rightarrow v \subseteq X. \end{cases}$$

Remark. For any deductively closed subset X of A we have $X \neq \varnothing \Leftrightarrow \Delta \in X$, since $u \vdash \Delta$ for any $u \in \boldsymbol{Fin}(A)$. We also call the elements of $\mathscr{D}(S)$ the *configurations* of S. For instance, given a non-empty subset X of A, the set

$$\bar{X} =_{\text{def}} \bigcup \{v \mid \exists u \subseteq X. u \vdash v\}$$

is an element of $\mathscr{D}(S)$, the least one containing X. Note that this element is also

$$\bar{X} = \bigcap \{U \mid U \in \mathscr{D}(S) \text{ and } X \subseteq U\}.$$

In this last formula, the subset X of A can be empty; we call this set $\bar{X}$ the *closure* of X. It can be noted that, for $u, v \in \boldsymbol{Fin}(A)$,

$$v \subseteq \bar{u} \Leftrightarrow u \vdash v.$$

Returning to our previous basic example, we can see that for a join semilattice $(K, \sqsubseteq)$ with bottom $\perp$, the elements of the domain of the information structure $(K, \perp, \vdash)$—as defined above—are the *ideals* of the poset $(K, \sqsubseteq)$. More generally, information structures provide us with a concrete presentation of algebraic lattices:

PROPOSITION. *A poset $(L, \sqsubseteq)$ is an algebraic lattice if and only if it is isomorphic to the domain of an information structure. More precisely:*

(i) *for any information system the poset* $(\mathscr{D}(S), \subseteq)$ *is an algebraic lattice, whose compact elements are the sets* $\bar{u}$ *for* $u \in \textbf{\textit{Fin}}(A)$;

(ii) *any algebraic lattice* $(L, \sqsubseteq)$ *is isomorphic to the family of ideals of the join semilattice with bottom of its compact elements* $(\mathscr{K}(L), \sqsubseteq)$.

In particular, it is easy to see that for all $X \in \mathscr{D}(S)$,

$$X = \bigcup_{u \subseteq X} \bar{u}.$$

In $\mathscr{D}(S)$, we can describe the join as follows:

$$X \sqcup Y = \overline{X \cup Y}.$$

We mentioned that the continuous functions between complete lattices form a complete lattice. This lattice is algebraic (resp. prime algebraic) if the domain and range and algebraic (resp. prime algebraic). Then we can give an explicit description of the continuous functions, at the level of information systems: a continuous function f is represented by an *approximable mapping* between information systems, which is the set of pairs (u, v) such that $(\bar{u}, \bar{v}) \in \text{graph}(f)$. This set has to satisfy some properties, listed in the following definition, adapted from Scott [43] (see also Coppo *et al.* [13]):

DEFINITION (Approximable Mappings). Let $S_i = (A_i, \Delta_i, \vdash_i)$ for $i = 0, 1$ be two information systems. An approximable mapping from S_0 to S_1 is a relation f from $\textbf{\textit{Fin}}(A_0)$ to $\textbf{\textit{Fin}}(A_1)$ such that:

(i) $\{\Delta_0\} f \{\Delta_1\}$

(ii) ufv and $ufw \Rightarrow uf(v \cup w)$

(iii) $u' \vdash_0 u$ and ufv and $v \vdash_1 v' \Rightarrow u'fv'$.

An approximable mapping f from S_0 to S_1 represents a function $\text{fun}(f): \mathscr{D}(S_0) \to \mathscr{D}(S_1)$ given by

$$\text{fun}(f)(X) =_{\text{def}} \bigcup \{v \mid \exists u \subseteq X. ufv\}.$$

This is more or less Plotkin's set-theoretical definition of application (see Barendregt [5]).

For instance, it is easy to see that, given $S = (A, \Delta, \vdash)$, the relation $\vdash$ is an approximable mapping over S, and that $\text{fun}(\vdash)$ is the identity function. Remark that for any approximable mapping f we have

$$\text{fun}(f)(\bar{u}) = \bigcup \{v \mid ufv\}.$$

The approximable mappings concretely represent continuous functions:

PROPOSITION. *If f is an approximable mapping from S_0 to S_1 then* $\mathrm{fun}(f)$ *is a continuous function from $\mathscr{D}(S_0)$ to $\mathscr{D}(S_1)$. Conversely, if g is a continuous function from $\mathscr{D}(S_0)$ to $\mathscr{D}(S_1)$ then the relation* $\mathrm{map}(g)$ *given by*

$$(u, v) \in \mathrm{map}(g) \Leftrightarrow_{\mathrm{def}} v \subseteq g(\bar{u})$$

is an approximable mapping such that $g = \mathrm{fun}(\mathrm{map}(g))$. This establishes an isomorphism between the poset of continuous functions, extensionally ordered, and the family of approximable mappings, ordered by inclusion.

3.1.3. *Construction of the Canonical Domains*

Now we use the information systems to solve the domain equations $D = (D \to D)_\perp$ and $D = V_\perp$, where $V = (V \to D)$. We have to give two constructions on information systems, respectively representing the lifting construct and the exponentiation—or more accurately the family of approximable mappings.

It is very easy to define the lifting of information structures: let $S = (A, \Delta, \vdash)$ be an information structure, and let us define S' as $(A', \Delta', \vdash')$, where $A' = A \cup \{\Delta'\}$ (with $\Delta' \notin A$), and $\vdash'$ is the least relation containing $\vdash$ and satisfying D1–D4 of Definition 3.1. Then it is not difficult to see that $\mathscr{D}(S')$ is isomorphic to $(\mathscr{D}(S))_\perp$, and more precisely

$$X \in \mathscr{D}(S') \Leftrightarrow X = \{\Delta'\} \text{ or } \exists Y \in \mathscr{D}(S).\, X = \{\Delta'\} \cup Y.$$

The second construct is the "exponentiation": let $S_i = (A_i, \Delta_i, \vdash_i)$ for $i = 0, 1$ be two information systems. We define

$$(S_0 \to S_1) = (A, \Delta, \vdash)$$

as follows:

(1) A is the set of pairs of finite non-empty subsets of A_0 and A_1; that is, $A = \boldsymbol{Fin}(A_0) \times \boldsymbol{Fin}(A_1)$

(2) $\Delta = (\{\Delta_0\}, \{\Delta_1\})$

(3) $\vdash$ is the least relation satisfying D1–D4 of Definition 3.1, and

(3.1) $(u, v), (u, w) \vdash (u, v \cup w)$

(3.2) $u' \vdash_0 u$ and $v \vdash_1 v' \Rightarrow (u, v) \vdash (u', v')$.

Note that due to clause D1 of Definition 3.1 we have

$$(u_1, v_1), \ldots, (u_n, v_n) \vdash (\{\Delta_0\}, \{\Delta_1\}).$$

Then we can show that this is an approximate representation of the domain of approximable mappings:

LEMMA. *$f \in \mathscr{D}(S_0 \to S_1)$ if and only if f is an approximable mapping from S_0 to S_1.*

A consequence of this result is that the poset $(\mathscr{D}(S_0) \to \mathscr{D}(S_1), \sqsubseteq)$ of continuous functions is isomorphic to $(\mathscr{D}(S_0 \to S_1), \subseteq)$. We are now ready to give the information systems describing the canonical domains; the constructions are very similar to Plotkin, Scott, and Engeler's (see Barendregt [5, Definition 5.4.5]). Intuitively the information system for $\boldsymbol{D}_\star$ is the least one that is closed for both lifting and exponentiation:

DEFINITION 3.2 (The Canonical Information System). The canonical information system is $S_\star = (A_\star, \Delta, \vdash_\star)$ where $A_\star$ is the least set such that

(i) $\Delta \in A_\star$

(ii) $u, v \in \boldsymbol{Fin}(A_\star) \Rightarrow (u, v) \in A_\star$

and $\vdash_\star$ is the least relation on $\boldsymbol{Fin}(A_\star)$ satisfying clauses D1–D4 of Definition 3.1, and also the clauses defining the exponentiation of information systems, that is,

$D_\star 5$. $u \in \boldsymbol{Fin}(A_\star - \{\Delta\}) \Rightarrow u \vdash_\star (\{\Delta\}, \{\Delta\})$

$D_\star 6$. $(u, v), (u, w) \vdash_\star (u, v \cup w)$

$D_\star 7$. $u' \vdash_\star u$ and $v \vdash_\star v' \Rightarrow (u, v) \vdash_\star (u', v')$.

Clearly if we let $\boldsymbol{D}_\star = \mathscr{D}(S_\star)$ then this domain satisfies the equation

$$\boldsymbol{D}_\star = (\boldsymbol{D}_\star \to \boldsymbol{D}_\star)_\perp .$$

For the strict canonical domain $\boldsymbol{D}_s$, the definition is quite similar:

DEFINITION 3.3 (The Strict Canonical Information System). The strict canonical information system is $S_s = (A_s \cup \{\Delta\}, \Delta, \vdash_s)$, where A_s is the least set such that

(i) $\nabla \in A_s$ with $\nabla \neq \Delta$

(ii) $u \in \boldsymbol{Fin}(A_s)$ and $v \in \boldsymbol{Fin}(A_s \cup \{\Delta\}) \Rightarrow (u, v) \in A_s$

and $\vdash_s$ is the least relation on $\boldsymbol{Fin}(A_s) \cup \{\Delta\}$ satisfying clauses D1–D4 of Definition 3.1, and also:

$D_s 5$. $u \in \boldsymbol{Fin}(A_s) \Rightarrow u \vdash_s \{\nabla\}$ and $u \vdash_s (\{\nabla\}, \{\Delta\})$

$D_s 6$. $(u, v), (u, w) \vdash_s (u, v \cup w)$

$D_s 7$. $u' \vdash_s u$ and $v \vdash_s v' \Rightarrow (u, v) \vdash_s (u', v')$

Note that in $D_s 6$ and $D_s 7$ it is simplicity assumed that $u, u' \in \boldsymbol{Fin}(A_s)$—

otherwise we could not form $(u, v \cup w)$, (u, v) and (u', v'). Clearly if we let $\boldsymbol{D}_s = \mathscr{D}(S_s)$ and $\boldsymbol{V} = \boldsymbol{D}_s - \{\Delta\}$ then these domains satisfy the equations:

$$\boldsymbol{D}_s = \boldsymbol{V}_\perp$$
$$\boldsymbol{V} = (\boldsymbol{V} \to \boldsymbol{D}_s)$$

3.2. *Logical Presentation of the Domains*

Information systems provide us with a concrete representation for algebraic lattices. In fact one can give an even more concrete presentation by using a *syntax* for information systems. This syntax will be a *logical* one: this formalizes Scott's intuitive explanations about information systems (cf. Coppo *et al.* [13], Dezani-Ciancaglini and Margaria [17], and Abramsky [2]).

The logical representation of algebraic lattices relies upon a statement *dual* to the one we used in introducing information systems, namely: *every algebraic lattice is isomorphic to the family of filters of a meet-semilattice with top element.* A meet-semilattice with top element $\top$ is a poset $(K, \sqsubseteq)$ such that any finite subset Y has a meet, that is, a greatest lower bound $\bigsqcap Y$. A filter is a subset X which is filtered (that is, for any finite subset Y of X there exists $x \in X$ such that $x \sqsubseteq \bigsqcap Y$) and satisfies $x \in X$ and $x \sqsubseteq y \Rightarrow y \in X$. Clearly a meet-semilattice $(K, \sqsubseteq)$ with top $\top$ determines an information structure $(K, \vdash, \top)$, where $\vdash$ is given by

$$a_1, \ldots, a_n \vdash b_1, \ldots, b_k \Leftrightarrow (a_1 \sqcap \cdots \sqcap a_n) \sqsubseteq (b_1 \sqcap \cdots \sqcap b_k).$$

The domain of this information structure is the family of filters of $(K, \sqsubseteq)$.

The formulae of the logical language associated with a given information system $S = (A, \Delta, \vdash)$ describe the finite non-empty subsets of A, that is, elements of $\boldsymbol{Fin}(A)$. We assume as given an atomic proposition for each singleton $\{a\}$, where $a \in A$. In particular we always use ω to denote $\{\Delta\}$, so that ω means "true." The union of two elements u and v of $\boldsymbol{Fin}(A)$ respectively denoted by ϕ and ψ is denoted by $\phi \wedge \psi$, the conjunction of ϕ and ψ. This is consistent with the interpretation of $u \cup v$ as a set of propositions that can be made about some elements of the Domain $\mathscr{D}(S)$. Then the logical language associated with S is the set of formulae given by the grammar

$$\phi ::= p \mid \omega \mid (\phi \wedge \phi),$$

where p is any atomic proposition.

The next step is to represent the deduction relation $\vdash$. This is turned into an *entailment* relation between formulae, $\phi \leqslant \psi$ (which could also be

written $\phi \Rightarrow \psi$, that is, "ψ is a consequence of ϕ") corresponding to $u \vdash v$. Then for instance a deduction

$$a_1, \ldots, a_n \vdash b_1, \ldots, b_k$$

is denoted by

$$p_1 \wedge \cdots \wedge p_n \leqslant q_1 \wedge \cdots \wedge q_k.$$

Translating clauses D1–D4 of Definition 3.1, we see that the entailment relation has to satisfy first the following two axioms corresponding to D1–D2:

$$\text{E1:}\quad \phi \leqslant \omega \qquad \text{E2:}\quad \frac{\phi \leqslant \tau,\ \tau \leqslant \psi}{\phi \leqslant \psi}.$$

To formulate D3 we remark that this requirement is equivalent to $u \vdash u$ and $u \cup v \vdash u$ (for $u, v \in \mathbf{Fin}(A)$). Since the conjunction is not a priori associative and commutative, we give two axioms for this last clause. Then we get

$$\text{E3.1:}\quad \phi \leqslant \phi \qquad \text{E3.2:}\quad (\phi \wedge \psi) \leqslant \phi \qquad \text{E3.3:}\quad (\phi \wedge \psi) \leqslant \psi.$$

Then $\leqslant$ is a preorder on formulae, and ω is the greatest, or "most general" formula. Finally the axiomatization of D4 is

$$\text{E4:}\quad \frac{\phi \leqslant \psi_0,\ \phi \leqslant \psi_1}{\phi \leqslant (\psi_0 \wedge \psi_1)}.$$

It we let

$$\phi \sim \psi \Leftrightarrow_{\text{def}} \phi \leqslant \psi \qquad \text{and} \qquad \psi \leqslant \phi$$

it is easy to see that

$$\phi \wedge \omega \sim \phi$$

$$\phi \wedge \psi \sim \psi \wedge \phi$$

$$\phi \wedge (\psi \wedge \zeta) \sim (\phi \wedge \psi) \wedge \zeta.$$

This allows us to use the notation $\bigwedge_i \phi_i$ for finite conjuctions, up to $\sim$ (with the convention that the empty conjuction is ω). One can also remark for instance that $\phi \sim (\phi \wedge \phi)$.

Remark 3.4. $\phi_0 \leqslant \psi_0$ and $\phi_1 \leqslant \psi_1 \Rightarrow \phi_0 \wedge \phi_1 \leqslant \psi_0 \wedge \phi_1$.

Finally we have to reflect in the logic the notion of element of the domain, that is, deductively closed non-empty subset of A. As we saw above, the appropriate notion is that of a filter of formulae:

DEFINITION (Filters). A filter is a set F of formulae such that

(i) $\omega \in F$

(ii) $\phi \in F$ and $\psi \in F \Rightarrow \phi \wedge \psi \in F$

(iii) $\phi \in F$ and $\phi \leqslant \psi \Rightarrow \psi \in F$.

For instance, for any formula ϕ the set $\phi\uparrow = \{\psi \mid \phi \leqslant \psi\}$ is a (principal) filter, generated by ϕ. We let the reader convinces him/herself that given an information system one can introduce some further axioms for entailment, so that the domain determined by the information system is isomorphic to the family of filters of this logic. We shall only detail the cases of $S_\star$ and S_s.

To deal with the exponentiation construct, we have to give a syntax for the finite non-empty sets of pairs (u, v)—in the strict case we also have to add a new propositional constant γ denoting $\{\nabla\}$. Taking advantage of the recursive nature of $A_\star$ and A_s, we can describe these pairs (or more precisely the singletons consisting of one of these pairs) as *implicative* formulae $(\phi \to \psi)$, and in the strict case $(\zeta \to \phi)$ where ζ and ϕ respectively represent elements of $\boldsymbol{Fin}(A_s)$ and $\boldsymbol{Fin}(A_s \cup \{\Delta\})$. An implicative formula $\phi \to \psi$ represents an element of information about a function f, meaning that if we have the information ϕ on the argument x, then we can be sure that the information ψ is true of the result $f(x)$. Then the logical language for $\boldsymbol{D}_\star$ is as follows:

$$(\Phi_\star) \quad \phi ::= \omega \mid (\phi \to \phi) \mid (\phi \wedge \phi).$$

As for $\boldsymbol{D}_s$, we have

$$(\Phi_s) \quad \psi ::= \omega \mid \zeta \mid (\psi \wedge \psi)$$

$$(\Gamma) \quad \zeta ::= \gamma \mid (\zeta \wedge \zeta) \mid (\zeta \to \psi).$$

The proposition γ could be called the "existence proposition," or the "convergence formula," a name that will be justified later.

We denote by $\leqslant_\star$ and $\leqslant_s$ the entailment relations respectively representing $\vdash_\star$ and $\vdash_s$. Both these relations have to satisfy clauses E1–E4. Then $\leqslant_\star$ is the last relation on $\Phi_\star$ satisfying E1–E4 (where one should obviously replace $\leqslant$ by $\leqslant_\star$), and the clauses corresponding to $D_\star 5$–$D_\star 7$, that is,

$$E_\star 5: \quad (\phi \to \omega) \leqslant_\star (\omega \to \omega)$$

$$E_\star 6: \quad ((\phi \to \phi_0) \wedge (\phi \to \phi_1)) \leqslant_\star (\phi \to (\phi_0 \wedge \phi_1))$$

$$E_\star 7: \quad \frac{\phi_0 \geqslant_\star \phi_1,\ \psi_0 \leqslant_\star \psi_1}{(\phi_0 \to \psi_0) \leqslant_\star (\phi_1 \to \psi_1)}.$$

Note that $E_\star 5$ is not the exact image of $D_\star 5$, which would be

$$(\phi_1 \to \psi_1) \wedge \cdots \wedge (\phi_n \to \psi_n) \leqslant_\star (\omega \to \omega)$$

However, it should be clear that this holds, since

$$\begin{aligned} (\phi_1 \to \psi_1) \wedge \cdots \wedge (\phi_n \to \psi_n) &\leqslant_\star (\phi_1 \to \psi_1) && \text{by E3.2} \\ &\leqslant_\star (\phi_1 \to \omega) && \text{by E1, } E_\star 7 \\ &\leqslant_\star (\omega \to \omega) && \text{by } E_\star 5 \end{aligned}$$

For what regards the strict case, we define the strict entailment relation $\phi \leqslant_s \psi$ as the least one satisfying E1–E4 above (where one should obviously replace $\leqslant$ by $\leqslant_s$), and the clauses corresponding to D_s5–D_s7, that is, using ζ, ζ_0, ζ_1 and ϕ_0, ϕ_1 to range over Γ and Φ_s respectively,

$$E_s 5.1: \quad \zeta \leqslant_s \gamma \qquad E_s 5.2: \quad \zeta \leqslant_s (\gamma \to \omega)$$

$$E_s 6: \quad ((\zeta \to \phi_0) \wedge (\zeta \to \phi_1)) \leqslant_s (\zeta \to (\phi_0 \wedge \phi_1))$$

$$E_s 7: \quad \frac{\zeta_0 \geqslant_s \zeta_1,\ \phi_0 \leqslant_s \phi_1}{(\zeta_0 \to \phi_0) \leqslant_s (\zeta_1 \to \phi_1)}.$$

One can see for instance that $\gamma \sim_s \gamma \to \omega$. This expresses the fact that there is just one least element in the domain V, namely $\gamma\uparrow$, and that the "propositions" $\{\nabla\}$ and $(\{\nabla\}, \{\Delta\})$ should be regarded as identical. More generally we have $\gamma \sim_s \zeta \to \omega$ for any $\zeta \in \Gamma$.

We could call $(\Gamma, \leqslant_s)$ the *logic of partial functions*. The logic $(\Phi_\star, \leqslant_\star)$ is exactly that of Dezani-Ciancaglini and Margaria [17] (if we forget about the propositional variables, which are not needed since there is a non-trivial initial solution to the canonical domain equation) and of Abramsky [3]. As a matter of fact, the "logics" $(\Phi_\star, \leqslant_\star)$ and $(\Phi_s, \leqslant_s)$ would perhaps better be named, following Curry, *functionality theories*, and the formulae *functional characters* (see also Coppo *et al.* [10]), since some "logical" properties do not hold here. For instance, although ω means "true" and $\phi \leqslant \psi$ means "ψ is a consequence of ϕ" (or ψ is more general than ϕ), one has $\phi \to \omega \not\sim_\star \omega$, and in fact "$\phi$ implies true" is strictly more informative than "true;" indeed one has $\lambda x \perp \neq \perp$. Similarly $\omega \to \phi \not\sim_\star \phi$. Note that $\phi \to \omega \sim \omega$ holds in the logic of Scott's D_∞ domain; see [12]. To

emphasize the fact that $\phi \to \psi$ is not the usual propositional implication, we could have used Abramsky's notation $(\phi \to \psi)_\perp$.

Let us denote by $\mathscr{F}(\Phi_\star, \leqslant_\star)$ and $\mathscr{F}(\Phi_s, \leqslant_s)$ the set of filters of our logics. It should be clear that they provide us with another representation of the canonical domains (cf. Coppo *et al.* [12], Abramsky [2, 3]:

PROPOSITION. *The canonical domain $\boldsymbol{D}_\star$ is isomorphic to the family $\mathscr{F}(\Phi_\star, \leqslant_\star)$, whose compact points are the principal filters $\phi\uparrow$ for $\phi \in \Phi_\star$. Similarly the strict canonical domain $\boldsymbol{D}_s$ is isomorphic to the family $\mathscr{F}(\Phi_s, \leqslant_s)$ of filters of the strict logic.*

Proof (Outline). We just treat the case of $\boldsymbol{D}_s$. One formalizes the idea that the formulae represent the finite non-empty sets of "propositions" of the strict canonical information system by defining $j: \Phi_s \to \boldsymbol{Fin}(A_s \cup \{\Delta\})$ as follows:

$$j(\omega) = \{\Delta\}$$

$$j(\phi \wedge \psi) = j(\phi) \cup j(\psi)$$

$$j(\gamma) = \{\nabla\}$$

$$j(\zeta \to \phi) = \{(j(\zeta), j(\phi))\}.$$

Then one proves that

(1) For any $a \in A_s \cup \{\Delta\}$ there exists $\phi \in \Phi_s$ such that $j(\phi) = \{a\}$ (this is very easy). Then for any $u \in \boldsymbol{Fin}(A_s) \cup \{\Delta\}$ there exists $\phi \in \Phi_s$ such that $j(\phi) = u$. Moreover $\phi \in \Gamma \Leftrightarrow j(\phi) \subseteq A_s$.

(2) If $\phi \leqslant_s \psi$ then $j(\phi) \vdash_s j(\psi)$ (by induction on the definition of $\phi \leqslant_s \psi$). Therefore if we let $h(X) = \{\phi \mid j(\phi) \subseteq X\}$ for $X \subseteq A_s \cup \{\Delta\}$ we have $X \in \boldsymbol{D}_s \Rightarrow h(X) \in \mathscr{F}(\Phi_s, \leqslant_s)$.

(3) If $j(\phi) \vdash_s j(\psi)$ then $\phi \leqslant_s \psi$, by induction on the definition of $j(\phi) \vdash_s j(\psi)$. This is a little bit more elaborate: one proves that $j(\psi) = u \cup v$ if and only if $\psi \sim_s \psi_0 \wedge \psi_1$ where $j(\psi_0) = u$ and $j(\psi_1) = v$. Therefore if we let $g(F) = \bigcup \{j(\phi) \mid \phi \in F\}$ for $F \subseteq \Phi_s$ we have $F \in \mathscr{F}(\Phi_s, \leqslant_s) \Rightarrow g(F) \in \boldsymbol{D}_s$.

Finally it is easy to check that the pair h, g provides us with the required isomorphism. ∎

3.3. *The Logical Systems*

In this section we give the interpretations of the λ_j^{nv}-calculus and λ_j^{v}-calculus in their respective canonical domains, that is $\boldsymbol{D}_\star$ and $\boldsymbol{D}_s$. We begin with an abstract definition of the semantics, and then derive their logical formulations, that is natural deduction systems, using as an intermediary step a concrete formulation in terms of approximable mappings. We are not very formal in this derivation, since we regard the natural

deduction systems as providing the actual semantics of the λ_j^{nv}-terms and λ_j^{v}-terms.

3.3.1. *Abstract Interpretation of the λ_j^{nv}- and λ_j^{v}-Calculi*

Let us recall that, following Scott's work (see Barendregt [5]), we can interpret the λ-calculus in any reflexive domain, that is, any domain D whose own function space is a retract of itself. This is the case for our canonical domains. In the strict case we take as "function space" the domain of strict continuous functions, since $\boldsymbol{D}_s$ is isomorphic to $(\boldsymbol{D}_s \to_\perp \boldsymbol{D}_s)_\perp$. In fact the specific retracts we use are given by the two obvious mappings associated with the lifting construct,

$$\mathrm{down}_\star : \boldsymbol{D}_\star \to (\boldsymbol{D}_\star \to \boldsymbol{D}_\star) \qquad \text{and} \qquad \mathrm{up}_\star : (\boldsymbol{D}_\star \to \boldsymbol{D}_\star) \to \boldsymbol{D}_\star$$

and

$$\mathrm{down}_s : \boldsymbol{D}_s \to \boldsymbol{V} \qquad \text{and} \qquad \mathrm{up}_s : \boldsymbol{V} \to \boldsymbol{D}_s,$$

satisfying $\mathrm{down}(\mathrm{up}(f)) = f$. Then we can interpret our calculi as usual, the semantic value $[\![M]\!]$ of the term $M \in \Lambda_j^{nv}$ being a mapping from *environments to* $\boldsymbol{D}_\star$, and similarly for $M \in \Lambda_j^{v}$. Given a domain D, an environment is a mapping $\rho : \boldsymbol{X} \to D$ assigning values to the variables, such that $\rho(x) = \perp$ except for finitely many variables. To give this interpretation we use the *updating* operation on environments, defined as follows: the updating of ρ by $e \in D$ at $x \in \boldsymbol{X}$, denoted $[x \mapsto e]\,\rho$, is given by

$$([x \mapsto e]\,\rho)(y) = \begin{cases} e & \text{if } \ y = x \\ \rho(y) & \text{otherwise.} \end{cases}$$

We also interpret a substitution $\sigma \in \Sigma$ as a mapping $[\![\sigma]\!]$ from environments to environments. Then, omitting the obvious subscript $*$ and s, the semantics of our calculi is given by the following equations:

$$[\![x]\!]\,\rho = \rho(x)$$

$$[\![\langle \lambda x M, \sigma \rangle]\!]_\star\,\rho = \mathrm{up}_\star(f) \quad \text{where} \quad f(e) = [\![M]\!]_\star\,([x \mapsto e]([\![\sigma]\!]_\star\,\rho))$$

$$[\![\langle \lambda^v x M, \sigma \rangle]\!]\,\rho = \mathrm{up}(g)$$

$$\text{where} \quad g(e) = \begin{cases} \perp & \text{if } \ e = \perp \\ [\![M]\!]([x \mapsto e]([\![\sigma]\!]\,\rho)) & \text{otherwise} \end{cases}$$

$$[\![MN]\!]\,\rho = (\mathrm{down}([\![M]\!]\,\rho))([\![N]\!]\,\rho)$$

$$[\![M \parallel N]\!]\,\rho = ([\![M]\!]\,\rho) \sqcup ([\![N]\!]\,\rho)$$

$$[\![\varepsilon]\!]\,\rho = \rho$$

$$[\![[M/x]\,\sigma]\!]\,\rho = [x \mapsto [\![M]\!]\,\rho]([\![\sigma]\!]\,\rho).$$

The extensional semantic preorders $M \sqsubseteq_\star N$ on λ_j^{nv}-terms and $M \sqsubseteq_s N$ for the λ_j^{v}-terms are given by the same definition, where D stands for $\boldsymbol{D}_\star$ or $\boldsymbol{D}_s$:

$$M \sqsubseteq N \Leftrightarrow_{\text{def}} \forall \rho : \boldsymbol{X} \to D.\, [\![M]\!]\, \rho \subseteq [\![N]\!]\, \rho.$$

Now we want to give a more explicit description of the semantics, using the concrete presentations of the canonical domains. First we can rephrase the previous equations, taking into account the fact that $[\![M]\!]\, \rho$ is a subset of A (where A is $A_\star$ or, in the strict case, $A_s \cup \{\Delta\}$), and more precisely a configuration of S (where S is $S_\star$ or S_s in the strict case). In view of the logical interpretation, we in fact describe $[\![M]\!]\, \rho$ as a union of elements of $\boldsymbol{Fin}(A)$. The first item, that is, $[\![x]\!]\, \rho = \rho(x)$, can be pedantically formulated as follows:

S1. $[\![x]\!]\, \rho$ is the least subset of A such that $u \subseteq \rho(x) \Rightarrow u \subseteq [\![x]\!]\, \rho$ where, as before, u ranges over $\boldsymbol{Fin}(A)$.

The concrete formulations of the interpretation of the closures is a bit more elaborate. Let us first deal with $[\![\langle \lambda x M, \sigma \rangle]\!]_\star\, \rho = \text{up}_\star(f)$, where

$$f(e) = [\![M]\!]_\star\, ([x \mapsto e]([\![\sigma]\!]_\star\, \rho))$$

is, by construction, a continuous function from $\boldsymbol{D}_\star$ to itself, or more precisely the approximable mapping

$$\{(u, v) \mid v \subseteq [\![M]\!]_\star\, ([x \mapsto \bar{u}]([\![\sigma]\!]_\star\, \rho))\}.$$

Since $\text{up}_\star(X) = X \cup \{\Delta\}$ we can describe more concretely the interpretation of $\langle \lambda x M, \sigma \rangle$ in $\boldsymbol{D}_\star$ as follows:

S2.1. $[\![\langle \lambda x M, \sigma \rangle]\!]_\star\, \rho$ is the least configuration of $S_\star$ such that

$$v \subseteq [\![M]\!]_\star\, ([x \mapsto \bar{u}]([\![\sigma]\!]_\star\, \rho)) \Rightarrow (u, v) \in [\![\langle \lambda x M, \sigma \rangle]\!]_\star\, \rho.$$

Similarly for $[\![\langle \lambda^{\text{v}} x M, \sigma \rangle]\!]_\star\, \rho = \text{up}_\star(g)$, the strict function g is represented by the approximable mapping

$$\{\Delta\} \cup \{(u, v) \mid \bar{u} \neq \{\Delta\} \text{ and } v \subseteq [\![M]\!]_\star\, ([x \mapsto \bar{u}]([\![\sigma]\!]_\star\, \rho))\}.$$

It is not difficult to see that $\bar{u} \neq \{\Delta\} \Leftrightarrow u \vdash_\star (\{\Delta\}, \{\Delta\})$; therefore we can describe more concretely the interpretation of $\langle \lambda^{\text{v}} x M, \sigma \rangle$ in $\boldsymbol{D}_\star$ as follows:

S2.2. $[\![\langle \lambda^{\text{v}} x M, \sigma \rangle]\!]_\star\, \rho$ is the least configuration of $S_\star$ such that

$$u \vdash_\star (\{\Delta\}, \{\Delta\})$$

and

$$v \subseteq [\![M]\!]_\star ([x \mapsto \bar{u}]([\![\sigma]\!]_\star \rho)) \Rightarrow (u, v) \in [\![\langle \lambda^{\nu} xM, \sigma \rangle]\!]_\star \rho.$$

We let the reader convinces him/herself that the following is an appropriate formulation of the interpretation of $\langle \lambda^{\nu} xM, \sigma \rangle$ in $\boldsymbol{D}_s$:

S$_s$2. $[\![\langle \lambda^{\nu} xM, \sigma \rangle]\!]_s \rho$ is the least configuration of S_s such that

$$u \in \boldsymbol{Fin}(A_s) \text{ and } v \subseteq [\![M]\!]_s ([x \mapsto \bar{u}]([\![\sigma]\!]_s \rho)) \Rightarrow (u, v) \in [\![\langle \lambda^{\nu} xM, \sigma \rangle]\!]_s \rho.$$

To interpret an application MN we used down($[\![M]\!] \rho$), or more precisely the function represented by this element of the domain, that is fun(down($[\![M]\!] \rho$)). Then it is not difficult to check that we can rephrase the definition of $[\![MN]\!] \rho$ as follows:

S3. $[\![MN]\!] \rho$ is the least configuration of S such that

$$u \subseteq [\![N]\!] \rho \text{ and } (u, v) \in [\![M]\!] \rho \quad \Rightarrow \quad v \subseteq [\![MN]\!] \rho.$$

Note that in the strict case, it is implicitly assumed that $u \in \boldsymbol{Fin}(A_s)$, since otherwise we could not form (u, v). Finally the interpretation of $[\![M \parallel N]\!] \rho$, that is, the join of $[\![M]\!] \rho$ and $[\![N]\!] \rho$, is quite simple to formulate:

S4. $[\![M \parallel N]\!] \rho$ is the least configuration of S such that

$$u \subseteq [\![M]\!] \rho \text{ and } v \subseteq [\![N]\!] \rho \quad \Rightarrow \quad u \cup v \subseteq [\![M \parallel N]\!] \rho.$$

3.3.2. *Logical Interpretation of the $\lambda_j^{n\nu}$- and λ_j^{ν}-Calculi*

Now we want to take a further step, making explicit use of the fact that the elements of the canonical domains are filters of formulae. Then we describe $\phi \in [\![M]\!] \rho$, where ρ is now a mapping from X to the family of filters (either $\mathscr{F}(\Phi_\star, \leqslant_\star)$ or $\mathscr{F}(\Phi_s, \leqslant_s)$), by means of a "typing system," or more accurately a *logical natural deduction system* allowing us to prove *sequents*, that is, statements of the form $H \vdash M : \phi$. In fact we have to elaborate two logical systems, one for $H \vdash_\star M : \phi$, with $M \in \Lambda_j^{n\nu}$ and $\phi \in \Phi_\star$, and the other for $H \vdash_s M : \phi$, with $M \in \Lambda_j^{\nu}$ and $\phi \in \Phi_s$. Since they share some rules, we begin with some generalities, not distinguishing the two kinds of sequents.

In a sequent $H \vdash M : \phi$, H is a "logical environment." called a *hypothesis*, that is, a mapping $H : X \to \Phi$ (where Φ is either $\Phi_\star$ or Φ_s) such that $H(x) \sim \omega$ except for finitely many variables. Then a hypothesis H may be

represented as usual as a sequence $x_1 : \phi_1, ..., x_n : \phi_n$. In particular we write $\vdash M : \phi$ for $H \vdash M : \phi$ where $H(x) \sim \omega$ for all x. Therefore the natural deduction systems will allow us to prove facts of the form

$$x_1 : \phi_1, ..., x_n : \phi_n \vdash M : \phi, \qquad (*)$$

to be read "from the hypothesis that the values of the variables x_i satisfy the respective formulae ϕ_i we may infer that the expression M satisfies ϕ." In fact we also use the comma occuring in the hypothesis as a symbol for the "updating" operation on hypotheses. More precisely, we define the updating of H by ϕ at x, denoted $(x : \phi, H)$, as follows:

$$(x : \phi, H)(y) = \begin{cases} \phi & \text{if } \ y = x \\ H(y) & \text{otherwise.} \end{cases}$$

Then in the notation $x_1 : \phi_1, ..., x_n : \phi_n$ a given variable may occur several times—its actual value is the leftmost one in the sequence.

We shall say that an environment ρ, which assigns a filter of formulae to a variable, *satisfies* the hypothesis H if $H(x) \in \rho(x)$ for all $x \in X$. To define the logical systems we give the logical formulation of the previous semantic equations S1–S4 defining the interpretations of our calculi, guided by the idea that a sequent of the form $(*)$ above means

$$\phi_1 \in \rho(x_1), ..., \phi_n \in \rho(x_n) \Rightarrow \phi \in [\![M]\!]\, \rho, \qquad (**)$$

that is, $H \vdash M : \phi$ iff $\phi \in [\![M]\!]\, \rho$ for all environments ρ satisfying H. The first rules to state assert that $[\![M]\!]\, \rho$ is a filter; that is,

$$\text{L1:} \quad H \vdash M : \omega \qquad \text{L2:} \quad \frac{H \vdash M : \phi,\ H \vdash M : \psi}{H \vdash M : (\phi \wedge \psi)}$$

$$\text{L3:} \quad \frac{H \vdash M : \phi}{H \vdash M : \psi}\ \phi \leqslant \psi.$$

Note that the usual rules for elimination of conjunction are derived from L3 (using E3.2, E.3.3):

$$\frac{H \vdash M : \phi \wedge \psi}{H \vdash M : \phi} \qquad \frac{H \vdash M : \phi \wedge \psi}{H \vdash M : \psi}$$

Now we give the rules corresponding to term formation. For the variables we have the obvious rule, corresponding to S1:

$$\text{L4:} \quad x : \phi,\ H \vdash x : \phi.$$

To give the logical meaning of $\langle \lambda x M, \sigma \rangle$ (in the "λ_j^{nv}-logical system") we

introduce another kind of sequent, namely $H \vdash \sigma : G$, to be read: "from the hypothesis H we may infer that the substitution σ satisfies the hypothesis G." The rules for this type of sequent are given below.

Then the "typing" rule for the usual, call-by-name closures corresponding to the definition S2.1 of $[\![\langle \lambda x M, \sigma \rangle]\!]_\star \rho$ is

$$\text{L5.1:} \quad \frac{x : \psi,\ G \vdash_\star M : \phi,\ H \vdash_\star \sigma : G}{H \vdash_\star \langle \lambda x M, \sigma \rangle : (\psi \to \phi)}.$$

Obviously this rule is not a part of the "λ_j^v-logical system". In this system we only have to give a rule for the call-by-value closures $\langle \lambda^v x M, \sigma \rangle$. In fact the rule is exactly the same as the previous one, with the exception that to form $(\psi \to \phi)$ we must have $\psi \in \Gamma$. We leave this assumption implicit, simply using ζ as a generic name for formulae of Γ. Then the strict rule, corresponding to S_s2, is

$$\text{L}_s5: \quad \frac{x : \zeta,\ G \vdash_s M : \phi,\ H \vdash_s \sigma : G}{H \vdash_s \langle \lambda^v x M, \sigma \rangle : (\zeta \to \phi)}.$$

To formulate the "typing" rule for strict closures in the λ_j^{nv}-logical system, we have to translate the assumption $u \vdash_\star (\{\Delta\}, \{\Delta\})$ of S2.2 into logical words. Then we get

$$\text{L5.2:} \quad \frac{x : \psi,\ G \vdash_\star M : \phi,\ H \vdash_\star \sigma : G}{H \vdash_\star \langle \lambda^v x M, \sigma \rangle : (\psi \to \phi)}\ \psi \leqslant_\star (\omega \to \omega).$$

To state the rule for application, corresponding to the semantic equation S3 for $[\![MN]\!]\rho$, we use an assumption $H \vdash M : (\psi \to \phi)$, corresponding to $(u, v) \in [\![M]\!]\rho$. Then in the strict case, we should have to say that $\psi \in \Gamma$ (this means that, unlike Scott [41], we use a restricted form of "modus ponens"). As previously, this is left implicit, and we give the same rule for both logical systems, namely

$$\text{L6:} \quad \frac{H \vdash M : (\psi \to \phi),\ H \vdash N : \psi}{H \vdash (MN) : \phi}.$$

The rule for the concurrent join, corresponding to S4, is obvious:

$$\text{L7:} \quad \frac{H \vdash M : \phi,\ H \vdash N : \psi}{H \vdash (M \parallel N) : (\phi \wedge \psi)}.$$

Finally we have to give the rules for $H \vdash \sigma : G$, whose meaning is

$$H \vdash [N_1/x_1] \cdots [N_k/x_k]\, \varepsilon : G \Leftrightarrow \forall i.\, H \vdash N_i : G(x_i).$$

Note, however, that since ε is the identity, we should also require that for any variable x not in $\{x_k, ..., x_k\}$ the formula $G(x)$ is in the filter generated by $H(x)$; that is, $H(x) \leqslant G(x)$. Let us introduce a notation:

$$H \leqslant G \Leftrightarrow_{\text{def}} \forall x \in \boldsymbol{X}. H(x) \leqslant G(x).$$

Then the rules for $H \vdash \sigma : G$ are the following:

$$\text{S1:} \quad \frac{}{H \vdash \varepsilon : G} H \leqslant G \qquad \text{S2:} \quad \frac{H \vdash M : \phi,\ H \vdash \sigma : G}{H \vdash [M/x]\,\sigma : (x : \phi, G)}.$$

Summarizing, the λ_j^{nv}-logical system consists of all the rules above, except $L_s 5$. Obviously we should have written $\vdash_\star$ instead of $\vdash$ in the general rules L1–L4, L6, L7, S1, and S2. A similar remark applies for the λ_j^{v}-logical system, whose rules are the ones above, except L5.1 and L5.2. Moreover, in these systems the terms and formulae are respectively taken to be either in Λ_j^{nv} and $\Phi_\star$ or in Λ_j^{v} and Φ_s. In what follows, when we write $H \vdash M : \phi$, in most cases we intend that this sequent be provable by means of the previous rules.

Let us see some examples. Since $\lambda x M$ is a notation for $\langle \lambda x M, \varepsilon \rangle$ we can derive the usual rule for abstractions:

$$\frac{x : \psi,\ H \vdash_\star M : \phi}{H \vdash_\star \lambda x M : (\psi \to \phi)}.$$

Then the inference rules L4, L5.1, L6 are the usual rules in Curry's typing system, while L1–L3 are typical of Coppo's typing systems [6.12]. One can also see that the following two rules are derivable from L7, using L1, L3, and $\phi \wedge \omega \leqslant \phi$:

$$\frac{H \vdash M : \phi}{H \vdash (M \parallel N) : \phi} \qquad \frac{H \vdash N : \psi}{H \vdash (M \parallel N) : \psi}.$$

In fact, we could have used these two rules instead of L7.

Some comments about these systems are in order: we do not regard the natural deduction systems as *typing* systems, because the notion of *type* usually refers to an idea of *constraint*. Types were introduced to avoid paradoxical situations, and more precisely to rule out self-application. They are in common use in mathematics and in programming, where they provide a discipline which ensures robust programs. In this case typability does not imply convergence, whereas in any "Curry–Howard" typing system a typable term is strongly convergent (as well as its subterms). Although the relation between the logical system and the computational notion of convergence turns out to be, as usual, a central question (see for

instance Leivant [23] or Krivine [21]), it should be clear that the purpose of "typing" systems à la Coppo is not to restrict program formation, but rather to tell something logical about meaningful programs. Then we could say, following Leivant [23], that Coppo's systems implement an idea of "types as formulae" (but not "programs as proofs"), where $\psi \to \phi$ is an assertion about the functional character of a program, and conjunction is just a way to join such assertions.

As we said above, we regard the logical systems as providing the actual interpretation of the λ_j^{nv} and λ_j^{v} calculi in the canonical (logical) domains. In particular, we may now define the *semantical preorders* on the terms of the calculi as follows:

$$M \sqsubseteq_{\mathscr{S}} N \Leftrightarrow_{\text{def}} \forall H \ \forall \phi . H \vdash M : \phi \Rightarrow H \vdash N : \phi.$$

Obviously this definition should have been particularized into two cases, namely $\sqsubseteq^{\star}_{\mathscr{S}}$ and $\sqsubseteq^{s}_{\mathscr{S}}$, for each of the calculi. However, in most occasions we give just one statement, which takes exactly the same form for the calculi.

Since the rules of the natural deduction systems for proving $H \vdash M : \phi$ are either independent of the term M, or composing assertions on the subterms of M, it is not surprising that these preorders are *precongruences*:

LEMMA 3.5. $M \sqsubseteq_{\mathscr{S}} N \Rightarrow \forall C . C[M] \sqsubseteq_{\mathscr{S}} C[N]$.

Proof. By a tedious induction on the inference of $H \vdash C[M] : \phi$, and by case on the context C, omitted. ∎

We shall also prove later (see Section 4.3) that the semantical preorders are preserved by instantiation; that is,

LEMMA 3.6. $M \sqsubseteq_{\mathscr{S}} N \Rightarrow \forall \sigma . M\sigma \sqsubseteq_{\mathscr{S}} N\sigma$.

4. COMPLETENESS AND FULL ABSTRACTION

4.1. *Testing*

In this section we introduce some preorders on terms of our calculi, which are denoted $\sqsubseteq$, decorated with various subscripts. The associated equivalences are denoted $\simeq$, with the corresponding subscripts. Obviously each preorder has two versions, one for the λ_j^{nv}-calculus and the other for the λ_j^{v}-calculus. These preorders should be denoted $\sqsubseteq^{\star}$ and $\sqsubseteq^{s}$, decorated with the relevant subscript, but we mostly leave this distinction implicit.

The definition of most of our preorders relies upon a *operational criterion*

according to which a (closed) convergent term is "better" than any other one; that is,

$$M \prec N \Leftrightarrow_{\text{def}} M \Downarrow \Rightarrow N \Downarrow .$$

This is a preorder, but it is not a precongruence: for instance, in the λ_j^{bv}-calculus, we have $\mathbf{F} \prec \mathbf{I}$ but $\mathbf{F}\Omega \not\prec \mathbf{I}\Omega$, and in the λ_j^{v}-calculus, $\lambda^{\text{v}}xx \prec \lambda^{\text{v}}x\Omega$, but $(\lambda^{\text{v}}xx)\,\lambda^{\text{v}}x\Omega \not\prec (\lambda^{\text{v}}x\Omega)\,\lambda^{\text{v}}x\Omega$.

Our first preorder is the *testing* preorder, already mentioned in the introduction. The *tests* are simply pairs (C, σ) made of a context C—to be more precise, a λ_j^{nv}-context or a λ_j^{v}-context—and environment σ, assigning λ_j^{nv}-terms or λ_j^{v}-terms to the variables. The succes of a test (C, σ) applied to M is the convergence of $C[M\sigma]$. This only makes sense if $C[M\sigma]$ can be evaluated, that is, if it is a closed term. Then in the definition of the testing preorder, we have to require the tested terms to be closed by the test:

DEFINITION (Testing or Observation Preorder).

$$M \sqsubseteq_{\mathscr{C}} N \Leftrightarrow_{\text{def}} \forall\sigma\ \forall C \text{ closing } M\sigma \text{ and } N\sigma.\quad C[M\sigma] \prec C[N\sigma]$$

Our main goal is to show a full abstraction result for each of our calculi; that is, we want to prove that the testing preorders coincide with the semantical preorders:

$$M \sqsubseteq_{\mathscr{S}} N \Leftrightarrow M \sqsubseteq_{\mathscr{C}} N.$$

To this end, we have to relate the testing ability with the logical character of the abstract interpretations of our calculi. Since these interpretations are functional, it is not surprising that a property of *operational extensionality* (see Bloom [9]), or a *context lemma* (see Milner [28] and Berry *et al.* [7]), holds. That is, the testing ability may be restricted to *applicative* λ_j^{nv}*-tests* and *applicative* λ_j^{v}*-tests*; namely, for the first case,

$$A ::= [\,] \mid \langle \lambda xA, \sigma \rangle \mid \langle \lambda^{\text{v}}xA, \sigma \rangle \mid (AM),$$

where M is any closed λ_j^{nv}-term and σ any λ_j^{nv}-environment. The corresponding testing preorders, that is, *applicative testing preorders*, are denoted $\sqsubseteq_{\mathscr{A}}$. For instance, $\mathbf{T} \not\cong_{\mathscr{A}} \mathbf{F}$, since these two terms are distinguished by the test $[\,]\Omega\mathbf{I}$. Similarly the terms $\mathbf{I}$ and $\mathbf{A} = \lambda xy.xy$ are distinguished by $[\,]\Omega$. We let the reader find similar examples for the λ_j^{v}-calculus. It should be obvious that testing is stronger than applicative testing; that is,

$$M \sqsubseteq_{\mathscr{C}} N \Rightarrow M \sqsubseteq_{\mathscr{A}} N.$$

We see later the converse implication, showing that applicative tests are enough to provide us with the full power of testing.

In the rest of this section we give some alternative characterizations of applicative testing, as a trace preorder, and by means of a kind of "logical relation." More precisely, we prove that applicative testing is equivalent to an extensional preorder on terms—in fact we only prove one implication here; the other one is a consequence of the full abstraction result. We cannot define extensional equality simply as the least relation $\simeq$ such that (for closed terms)

$$M \simeq N \Leftrightarrow \forall R.\, MR \simeq NR,$$

since this relation is incosistent: $M \simeq N$ for all M and N is the least fixed-point of this recursive definition. We have to incorporate a type distinction, giving some information on the functional character of M and N. In fact this distinction is given by our operational criterion: a (closed) term is "functional" if it is convergent. The definition of the extensional preorder, due to Abramsky [3] and inspired by the well-known Park and Milner notion of bisimulation, is as follows:

DEFINITION AND LEMMA (Applicative Simulation and Extensional Preorder). *A relation $\mathscr{R}$ on closed term is an applicative simulation if*

$$M \mathscr{R} N \Rightarrow \begin{cases} M \prec N & \text{and} \\ \forall Q.\, (MQ)\, \mathscr{R}\, (NQ). \end{cases}$$

The relation on closed terms given by

$$M \sqsubseteq_{\varepsilon} N \Leftrightarrow_{\text{def}} \exists \mathscr{R}\ (\textit{applicative simulation})\ M \mathscr{R} N$$

is an applicative simulation and preorder, called the extensional preorder.

Proof. One shows that the composition of two applicative simulations is an applicative simulation. ∎

The ($\lambda_{\mathrm{j}}^{\mathrm{nv}}$ and $\lambda_{\mathrm{j}}^{\mathrm{v}}$) extensional preorders are extended to arbitrary terms by instantiation:

$$M \sqsubseteq_{\varepsilon} N \Leftrightarrow_{\text{def}} \forall \sigma \text{ closing } M \text{ and } N.\quad M\sigma \sqsubseteq_{\varepsilon} N\sigma.$$

An important example of applicative simulation is provided by the convertibility relation of Section 2.2:

LEMMA 4.1. *The convertibility relation $\bowtie$ is an applicative simulation; therefore,*

$$M \bowtie N \Rightarrow M \simeq_{\varepsilon} N.$$

This is a direct consequence of Proposition 2.12. Let us see some other examples. The first one shows that, for M and N closed,

$$M \sqsubseteq_{\varepsilon} (M \parallel N) \qquad \text{and} \qquad M \sqsubseteq_{\varepsilon} (N \parallel M).$$

LEMMA 4.2. *Let $\mathscr{R}$ be the relation on closed terms given by*

$$M \mathscr{R} N \Leftrightarrow_{\text{def}} \exists P.\quad N \bowtie (M \parallel P) \text{ or } N \bowtie (P \parallel M).$$

Then $\mathscr{R}$ is an applicative simulation.

Proof. Clearly $M \prec (M \parallel P)$ (by the evaluation rule R3.1) and $M \prec (P \parallel M)$ (by the evaluation rule R3.2), therefore $M \mathscr{R} N \Rightarrow M \prec N$ by the Corollary 2.11. If $M \mathscr{R} N$, with for instance $N \bowtie (M \parallel P)$, then $NQ \bowtie (M \parallel P)\, Q \bowtie (MQ \parallel PQ)$ by definition of the convertibility relation. Therfore $M \mathscr{R} N \Rightarrow MQ \mathscr{R} NQ$ for all closed term Q. ∎

Another example of applicative simulation shows that applicative testing is stronger than the extensional preorder:

LEMMA 4.3. *The relation $\mathscr{R}$ on closed terms given by*

$$M \mathscr{R} N \Leftrightarrow_{\text{def}} \forall A \qquad (\textit{closed applicative test}) \quad A[M] \prec A[N]$$

is an applicative simulation; therefore

$$M \sqsubseteq_{\mathscr{A}} N \Rightarrow M \sqsubseteq_{\varepsilon} N.$$

Proof. Clearly $M \mathscr{R} N \Rightarrow M \prec N$ (take $A = \square$ in the definition), and $M \mathscr{R} N \Rightarrow MQ \mathscr{R} NQ$ since for any applicative test A the context $A[\square Q]$ is an applicative test, and $A[MQ] = (A[\square Q])[M]$. Since $(M\sigma)\,\rho = M(\sigma \circ \rho)$ (Lemma 2.1) it is easy to see that

$$M \sqsubseteq_{\mathscr{A}} N \Rightarrow \forall \sigma . M\sigma \sqsubseteq_{\mathscr{A}} N\sigma.$$

Therefore if $M \sqsubseteq_{\mathscr{A}} N$ then $M\sigma \sqsubseteq_{\mathscr{A}} N\sigma$ for all σ closing M and N, hence $M\sigma \mathscr{R} N\sigma$, and $M \sqsubseteq_{\varepsilon} N$ by definition of the extensional preorder. ∎

The extensional preorder coincides with a trace preorder, which we define now:

DEFINITION (Trace Preorder). *The set $\mathscr{T}(M)$ of traces of a closed term M is the set of (possibly empty) sequences $R_1 \cdots R_k$ of closed terms such that $MR_1 \cdots R_k \Downarrow$. The trace preorder on closed terms is given by*

$$M \sqsubseteq_{\mathscr{T}} N \Leftrightarrow_{\text{def}} \mathscr{T}(M) \subseteq \mathscr{T}(N).$$

Obviously we should have defined λ_j^{nv}-traces of λ_j^{nv}-terms, and λ_j^{v}-traces of λ_j^{v}-terms. The trace preorder is extended to arbitrary terms by instantiation:

$$M \sqsubseteq_{\mathscr{T}} N \Leftrightarrow_{\text{def}} \forall \sigma \text{ closing } M \text{ and } N.\quad M\sigma \sqsubseteq_{\mathscr{T}} N\sigma.$$

Now we show that the trace preorder and extensional preorder coincide:

LEMMA 4.4. $M \sqsubseteq_{\varepsilon} N \Leftrightarrow M \sqsubseteq_{\mathscr{T}} N$.

Proof. Clearly the relation $\sqsubseteq_{\tau}$, on closed terms, is an applicative simulation, since $M \Downarrow$ if and only if the empty sequence is a trace of M, and $U \in \mathscr{T}(MQ) \Leftrightarrow QU \in \mathscr{T}(M)$. Then $M \sqsubseteq_{\tau} N \Rightarrow M \sqsubseteq_{\varepsilon} N$. Conversely, one easily proves by induction on k that if $\mathscr{R}$ is an applicative simulation then

$$M \mathscr{R} N \Rightarrow \forall R_1, \ldots, R_k . (MR_1 \cdots R_k)\, \mathscr{R} (NR_1 \cdots R_k).$$

Moreover $P \mathscr{R} Q \Rightarrow P \prec Q$ by definition, hence the lemma. ∎

Summarizing, we have established that

$$M \sqsubseteq_{\mathscr{C}} N \Rightarrow M \sqsubseteq_{\mathscr{A}} N \Rightarrow M \sqsubseteq_{\varepsilon} N \Leftrightarrow M \sqsubseteq_{\tau} N.$$

We shall see that, as a consequence of the full abstraction result, all these preorders are in fact the same.

4.2. *Realizability and Soundness*

A further step towards the full abstraction result consists in giving a "realizability" interpretation of the formulae of $\Phi_{\star}$ and Φ_{s}. That is, each formula denotes a set of closed terms (respectively λ_j^{nv}-terms or λ_j^{v}-terms) realizing the given formula. This interpretation is due, for the typed λ-calculus, to Tait. It was originally called "convertibility," and later on received the names of "reducibility" or "computability" (a generalization of this semantics of formulae is credited to Reynolds and Scott in [19, 6]). We denote "M realizes ϕ" by $\models M : \phi$. We have to define a λ_j^{nv}-realizability, denoted $\models_{\star} M : \phi$, and a λ_j^{v}-realizability, $\models_{s} M : \phi$. The first two clauses of the two definitions are the same:

$$\models M : \omega \Leftrightarrow_{\text{def}} \text{true}$$

$$\models M : (\phi \wedge \psi) \Leftrightarrow_{\text{def}} \models M : \phi \text{ and } \models M : \psi.$$

Then a closed term realizes the formula $\psi \to \phi$ when it is a functional term, that is, it converges, and when applied to (closed) arguments realizing ψ then any result it returns realizes ϕ:

$$\models M : (\psi \to \phi) \Leftrightarrow_{\text{def}} M \Downarrow \text{ and } \forall R \models R : \psi \Rightarrow \models MR : \phi.$$

Note that in the case of λ_j^{nv}-realizability the term R ranges over $(\Lambda_j^{nv})^\diamond$, while in the case of λ_j^{v}-realizability R belongs to $(\lambda_j^{v})^\diamond$. Moreover, in this last case, the formula ψ is an element of Γ. This interpretation suggests that we could have denoted $\psi \to \phi$ as a "Hennessy–Milner modal formula" $[\psi]\,\phi$. As one can see, this interpretation of the formulae of $\Phi_\star$, adapted from the one given by Abramsky in [3], is essentially the "F-semantics" of Hindley ([19]; see also Dezani-Ciancaglini and Margaria [17]). The realizability interpretation of $\Phi_\star$ relies entirely upon the convergence property $M \Downarrow$. In fact it entails a characterization of this property: a closed λ_j^{nv}-term is converent if and only if it realizes the formula $\omega \to \omega$, that is,

Remark 4.5. $\forall M \in (\Lambda_j^{nv})^\diamond \; M \Downarrow \Leftrightarrow \models_\star M : (\omega \to \omega)$.

In the strict case, the convergence formula is γ. Then the last clause of the definition of λ_j^{v}-realizability is

$$\models_s M : \gamma \Leftrightarrow_{\text{def}} M \Downarrow .$$

To simplify the presentation, we also denote by γ the convergence formula of the logic $\Phi_\star$, that is, $\gamma = \omega \to \omega$.

We can introduce a *logical preorder* (in fact, two preorders) on closed terms relative to this interpretation of the formulae: a term is logically less than another one whenever it realizes fewer formulae. That is,

$$M \sqsubseteq_{\mathscr{L}} N \Leftrightarrow_{\text{def}} \forall \phi . \models M : \phi \Rightarrow \models N : \phi .$$

This preorder is extended to arbitrary terms by instantiation:

$$M \sqsubseteq_{\mathscr{L}} N \Leftrightarrow_{\text{def}} \forall \sigma \text{ closed } M\sigma \sqsubseteq_{\mathscr{L}} N\sigma .$$

Now we want to show that this preorder is weaker than the testing preorder. By virtue of the results of the previous section, it is enough to show.

LEMMA. *If $\mathscr{R}$ is an applicative simulation and $M \mathscr{R} N$ then $\models M : \phi \Rightarrow \models N : \phi$ for any fomula ϕ,*

whose proof (by induction on the formula ϕ) is trivial. As a consequence, we have now

PROPOSITION 4.6.

$$M \sqsubseteq_{\mathscr{C}} N \Rightarrow M \sqsubseteq_{\mathscr{A}} N \Rightarrow M \sqsubseteq_{\mathscr{E}} N \Leftrightarrow M \sqsubseteq_{\mathscr{T}} N \Rightarrow M \sqsubseteq_{\mathscr{L}} N .$$

Now to show that the testing preorder is stronger than the semantical preorder, we have to relate the realizability interpretation of the logics with the natural deduction system. Indeed the comparison of $\models M : \phi$

and $\vdash M : \phi$ is the central topic of this work. As usual, this comparison takes the form of a *soundness and completeness* result. We prove first the soundness of the natural deduction systems with respect to an extension of the realizability interpretation. We let $H \models M : \phi$, to be read as "M realizes ϕ under the hypothesis H," if $M\sigma$ realizes ϕ for any substitution σ satisfying the hypothesis H regarding the free variables of M. Formally,

$$H \models M : \phi \Leftrightarrow_{\text{def}} \forall \sigma (\forall x \in \text{fv}(M).\, \models \sigma(x) : H(x)) \Rightarrow \models M\sigma : \phi.$$

Note that in this definition it is implicitly assumed that $\sigma(x)$ is a closed term for any $x \in \text{fv}(M)$.

Remark 4.7. $M \sqsubseteq_{\mathscr{L}} N \Rightarrow \forall \phi\, \forall H.\, H \models M : \phi \Rightarrow H \models N : \phi$.

The soundness property of the natural deduction systems is

$$H \vdash M : \phi \Rightarrow H \models M : \phi.$$

The argument is the standard one: this implication is proved by induction on the inference of the sequent $H \vdash M : \phi$. To this end we first show the validity of the natural deduction rules, apart the rules concerning the closures. The validity of L1 and L2 is trivial. The validity of L3 results from the following:

LEMMA 4.8. *If* $\models M : \phi$ *and* $\phi \leqslant \psi$ *then* $\models M : \psi$.

Proof. By induction on the definition of $\phi \leqslant \psi$, straightforward (in the case of $E_s 5.1$ and $E_s 5.2$, one proves $\zeta \in \Gamma$ and $\models M : \zeta \Rightarrow M \Downarrow$, by induction on the formula ζ). ∎

By definition of the interpretation of $(\psi \to \phi)$, the rule L6 is valid. It is easy to see that the rule L7 is valid: if $H \models M : \phi$ and $H \models N : \psi$, let σ be a substitution such that $\models \sigma(x) : H(x)$ for any $x \in \text{fv}(M \parallel N)$. Then $\models M\sigma : \phi$ and $\models N\sigma : \psi$. Since $M\sigma \sqsubseteq_{\mathscr{L}} (M\sigma \parallel N\sigma)$ and $N\sigma \sqsubseteq_{\mathscr{L}} (M\sigma \parallel N\sigma)$ by Lemma 4.2 and Proposition 4.6, we have $\models (M \parallel N)\sigma : \phi$ and $\models (M \parallel N)\sigma : \psi$, therefore $H \models (M \parallel N) : (\phi \wedge \psi)$.

PROPOSITION (Soundness). *If* $H \vdash M : \phi$ *then* $H \models M : \phi$.

Proof. In fact we prove simultaneously

$$H \vdash M : \phi \Rightarrow H \models M : \phi$$

and

$$H \vdash \sigma : G \Rightarrow \forall x.\, H \models \sigma(x) : G(x)$$

by induction on the inference of the sequents. From the previous remarks, we only have to examine the cases of S1, S2, L5.1, L_s5, and L5.2.

• If $H \vdash \varepsilon : G$ (by means of S1) with $H(x) \leqslant G(x)$ for any variable x, and if ρ is a substitution such that $\models \rho(x) : H(x)$, then $\models \rho(x) : G(x)$ by the previous lemma, and $x\rho = (\varepsilon(x))\ \rho = \rho(x)$, therefore $H \models \varepsilon(x) : G(x)$.

• If $H \vdash [M/x]\ \sigma : (x : \phi, G)$ by S2, that is, $H \vdash M : \phi$ and $H \vdash \sigma : G$, then one simply uses the induction hypotheses.

• If $M = \langle \lambda x P, \sigma \rangle$, $\phi = \psi \to \tau$, $x : \psi$, $G \vdash_\star P : \tau$ and $H \vdash_\star \sigma : G$, let ρ be a substitution such that $\models \rho(z) : H(z)$ for any $z \in \mathrm{fv}(M)$. We have to show that $\models M\rho : \psi \to \tau$. Clearly $M\rho \Downarrow$ since $M\rho$ is a closure. Let R be a closed term such that $\models R : \psi$. We have $(M\rho)\ R \bowtie P[R/x](\sigma \circ \rho)$; therefore, by Lemma 4.1 and Proposition 4.6, it is enough to show that $\models P[R/x](\sigma \circ \rho) : \tau$. By induction hypothesis on the inference of $H \vdash_\star \sigma : G$ we have $H \models \sigma(y) : G(y)$ for any variable y, therefore $\models (\sigma \circ \rho)(y) : G(y)$ for $y \in \mathrm{fv}(P) - \{x\}$. Then we use the induction hypotheses on the proof of $x : \psi,\ G \vdash_\star P : \tau$.

• The case of L_s5 is similar: here we have $M = \langle \lambda^{\curlyvee} x P, \sigma \rangle$ and $\phi = \zeta \to \tau$, with $\zeta \in \Gamma$. Clearly $\zeta \leqslant_s \gamma$ by $E_s5.1$; therefore if $\models_s R : \zeta$ then $R \Downarrow$. By Proposition 2.12 we have in this case $(M\rho)\ R \bowtie P[R/x](\sigma \circ \rho)$. The rest of the proof is the same as for L5.1. The proof in the case of L5.2 follows exactly the same lines. ▮

4.3. *Main Results: Completeness and Full Abstraction*

As announced, we prove a completeness theorem, that is,

$$H \models M : \phi \Rightarrow H \vdash M : \phi.$$

An obvious consequence of soundness and completeness (and of Proposition 4.6 and Remark 4.7) is a first half of the full abstraction result, namely that the testing preorders are stronger than the semantical preorders:

$$M \sqsubseteq_{\mathscr{C}} N \Rightarrow M \sqsubseteq_{\mathscr{S}} N.$$

Before seeing the converse, let us indicate what are the main points in proving the completeness theorem. Most of these auxiliary results are proved in the next section. First we show some structural properties, namely weakening, cut, and paste. Our weakening result is somewhat stronger than the usual one. It relies upon an extension of the entailment relation to hypotheses, relative to a given set of variables V. Let us define

$$G \leqslant^V H \Leftrightarrow_{\mathrm{def}} \forall x \in V.\, G(x) \leqslant H(x).$$

We omit the superscript when it is X. Then the *weakening* lemma (Lemma 5.2)—which would perhaps better be named "strenghtening"[3]—asserts that if under the hypothesis H we can prove that M satisfies ϕ, then by strengthening the hypothesis, we are able to prove the same fact. Formally, if $\mathrm{fv}(M) \subseteq V$ then

$$H \vdash M : \phi \text{ and } G \leqslant^{V} H \quad \Rightarrow \quad G \vdash M : \phi.$$

Then we have the usual *cut* property, that is, the validity of the rule

$$\frac{H \vdash N : \psi,\ x : \psi,\ H \vdash M : \phi}{H \vdash M[N/x]\,\varepsilon : \phi}.$$

We prove more generally (Proposition 5.5)

$$H \vdash \sigma : G \text{ and } G \vdash M : \phi \quad \Rightarrow \quad H \vdash M\sigma : \phi.$$

In Coppo's typing systems a kind of converse property, which we call *paste* (Proposition 5.10), also holds; see for instance Krivine [21]. This property says that a proof of $H \vdash M\sigma : \phi$ is always composed of proofs of intermediary assertions for the $\sigma(x)$'s and the proof that M satisfies ϕ under the corresponding hypothesis:

$$H \vdash M\sigma : \phi \quad \Rightarrow \quad \exists G.\, H \vdash \sigma : G \text{ and } G \vdash M : \phi.$$

The main property we have to prove is that the computational property of convergence can be ascertained using the natural deduction system:

CONVERGENCE LEMMA.

$$H \models M : \gamma \Rightarrow H \vdash M : \gamma.$$

To prove this lemma, we use two auxiliary results, the first one being (Corollary 5.14).

CONVERGENCE LEMMA, CLOSED TERMS. *If M is a closed term then $M \Downarrow \Rightarrow \vdash M : \gamma$.*

The crucial fact allowing us to prove the convergence lemma is the definability of the compact elements of the canonical domains, that is, the principal filters generated by a formula $\{\psi \mid \phi \leqslant \psi\}$. Then the definability result can be stated as follows (Corollary 5.22):

[3] The usual names of the structural properties refer to a bottom-up search for a proof of a given sequent. If we think of a top-down construction of proofs, we should find dual names for these properties, exchanging for instance cut and paste.

CHARACTERIZATION LEMMA. *For any formula $\phi \in \Phi_\star$ (resp. $\phi \in \Phi_s$) there exists a closed λ_j^{nv}-term (resp. λ_j^{v}-term) $\mathbf{M}_\phi$ such that*

$$\vdash \mathbf{M}_\phi : \psi \Leftrightarrow \phi \leqslant \psi.$$

This is the only result for which we need to have the call-by-value closures and the concurrent join in the syntax. The fact that these two ingredients are really needed is shown by Abramsky's and Ong non-full-abstraction results. To define the characteristic terms $\mathbf{M}_\phi$ we associate with each formula ϕ a "test for ϕ;" that is, in the case of the λ_j^{nv}-calculus, a closed λ_j^{nv}-term $\mathbf{T}_\phi$ such that $\models_\star M : \phi \Rightarrow \mathbf{T}_\phi M \Downarrow \mathbf{I}$. For the λ_j^{v}-calculus these terms are only defined for fomulae of Γ, and they are such that $\models_s M : \zeta \Rightarrow \mathbf{T}_\zeta M \Downarrow \mathbf{I}_s$, where the strict identity $\mathbf{I}_s$ is $\lambda^v xx$. The construction of these terms is, in the case of the λ_j^{nv}-calculus, as follows:

$$\begin{aligned}
\mathbf{M}_\omega &= \Omega \\
\mathbf{T}_\omega &= \mathbf{F} = \lambda y \lambda x x \\
\mathbf{M}_{\phi \wedge \psi} &= (\mathbf{M}_\phi \parallel \mathbf{M}_\psi) \\
\mathbf{T}_{\phi \wedge \psi} &= \lambda x (\mathbf{T}_\phi x)(\mathbf{T}_\psi x) \\
\mathbf{M}_{\phi \to \psi} &= \lambda x (\mathbf{T}_\phi x)\, \mathbf{M}_\psi \\
\mathbf{T}_{\phi \to \psi} &= \lambda^v x (\mathbf{T}_\psi (x \mathbf{M}_\phi)).
\end{aligned}$$

As for the λ_j^{v}-calculus, we define the characteristic terms and testing terms as follows:

$$\begin{aligned}
\mathbf{M}_\omega &= \Omega_s \\
\mathbf{M}_\gamma &= \lambda^v x \Omega_s \\
\mathbf{T}_\gamma &= \mathbf{F}_s = \lambda^v y \lambda^v x x \\
\mathbf{M}_{\phi \wedge \psi} &= (\mathbf{M}_\phi \parallel \mathbf{M}_\psi) \\
\mathbf{T}_{\zeta \wedge \tau} &= \lambda^v x (\mathbf{T}_\zeta x)(\mathbf{T}_\tau x) \\
\mathbf{M}_{\zeta \to \phi} &= \lambda^v x (\mathbf{T}_\zeta x)\, \mathbf{M}_\phi \\
\mathbf{T}_{\zeta \to \phi} &= \begin{cases} \mathbf{T}_\gamma & \text{if } \phi \in \Phi_s - \Gamma \\ \lambda^v x \mathbf{T}_\phi (x \mathbf{M}_\zeta) & \text{otherwise.} \end{cases}
\end{aligned}$$

We can now prove that the convergence lemma holds:

Proof of the Convergence Lemma. Assume that $H \models M : \gamma$, $\mathrm{fv}(M) = \{x_1, \ldots, x_n\}$, and $H(x_i) = \phi_i$. Let χ be the substitution given by

$$\chi = [\mathbf{M}_{\phi_1}/x_1] \cdots [\mathbf{M}_{\phi_n}/x_n]\, \varepsilon.$$

Then, since $\models \chi(x) : H(x)$ for $x \in \mathrm{fv}(M)$ by the characterization lemma above and soundness, we have $M\chi \Downarrow$; hence, by the convergence lemma for closed terms,

$$\vdash M\chi : \gamma.$$

Then by paste there exists $\psi_1, ..., \psi_n$ such that

$$x_1 : \psi_1, ..., x_k : \psi_n \vdash M : \gamma$$

and $\vdash \chi(x_i) : \psi_i$. Then, by the characterization lemma, $H(x_i) \leqslant \psi_i$ for $x_i \in \mathrm{fv}(M)$; therefore by weakening $H \vdash M : \gamma$. ∎

Finally, to prove the completeness theorem, we use Hindley's method [19, 20] by means of a refinement of the usual deduction theorem, namely (Lemma 5.7).

EXTENSIONALITY LEMMA (The Deduction Theorem).

$$H \vdash M : \phi \to \psi \Leftrightarrow H \vdash M : \gamma \text{ and } x : \phi,\ H \vdash Mx : \psi \quad \text{for some} \quad x \notin \mathrm{fv}(M).$$

Dezani-Ciancaglini and Margaria showed in [17] that such a property was needed for completeness: they added to the "typing" system a rule, called Hindley's rule, whose typical instance is

$$\frac{H \vdash M : \phi \wedge (\omega \to \omega)}{H \vdash \lambda x(Mx) : \phi} \qquad x \notin \mathrm{fv}(M).$$

Proof of the Completeness Theorem. We show that $H \models M : \phi \Rightarrow H \vdash M : \phi$ by induction on the formula ϕ. The cases where $\phi = \omega$ and $\phi = \phi_0 \wedge \phi_1$ are obvious (using L1 and L2). If $\phi = \gamma$ then we use the convergence lemma.

If $H \models M : \phi \to \psi$ then $\models M\sigma : \phi \to \psi$ for any substitution σ such that $\models \sigma(z) : H(z)$ for all $z \in \mathrm{fv}(M)$. Then $M\sigma \Downarrow$ and $\models (M\sigma)\, R : \psi$ for any closed term R such that $\models R : \phi$. Since $H \models M : \gamma$ we have $H \vdash M : \gamma$ by the convergence lemma. Let x be a variable not in $\mathrm{fv}(M)$. Then by Lemma 2.3 and Propositions 2.12 and 4.6, we have

$$(M\sigma)\, R \simeq_{\mathscr{L}} (Mx)[R/x]\, \sigma,$$

hence $x : \phi,\ H \models Mx : \psi$ by definition. By the induction hypothesis

$$x : \phi,\ H \vdash Mx : \psi;$$

therefore $H \vdash M : \phi \to \psi$ by the extensionality lemma. ∎

To conclude this section we show how to prove the second half of the full abstraction result, that is, more precisely, the *adequacy theorem*:

$$M \sqsubseteq_{\mathscr{S}} N \Rightarrow M \sqsubseteq_{\mathscr{C}} N.$$

We have to prove that $M \sqsubseteq_{\mathscr{S}} N \Rightarrow C[M\sigma] \prec C[N\sigma]$ for any substitution σ and context C, closing $M\sigma$ and $N\sigma$. We have seen (Lemma 3.5) that

$$M \sqsubseteq_{\mathscr{S}} N \Rightarrow \forall C.C[M] \sqsubseteq_{\mathscr{S}} C[N].$$

We also announced that the semantical preorders are preserved by instantiation (Lemma 3.6). We can now prove this result:

LEMMA. $M \sqsubseteq_{\mathscr{S}} N \Rightarrow \forall \sigma . M\sigma \sqsubseteq_{\mathscr{S}} N\sigma$.

Proof. If $H \vdash M\sigma : \phi$ then by paste there exists an hypothesis G such that $H \vdash \sigma : G$ and $G \vdash M : \phi$. Then $G \vdash N : \phi$ since $M \sqsubseteq_{\mathscr{S}} N$, and $H \vdash N\sigma : \phi$ by cut. ∎

An obvious consequence of these lemmas is

$$M \sqsubseteq_{\mathscr{S}} N \Rightarrow \forall \sigma \, \forall C. [M\sigma] \sqsubseteq_{\mathscr{S}} C[N\sigma].$$

Therefore to prove the full abstraction theorem it just remains to show that a closed term is convergent if and only if it has a non-trivial semantic value, that is:

LEMMA 4.9 (Computational Adequacy). *For any closed term* M

$$[\![M]\!] \neq \bot \Leftrightarrow M \Downarrow .$$

Proof. Since $[\![M]\!] \neq \bot \Leftrightarrow \gamma \in [\![M]\!] \Leftrightarrow \vdash M : \gamma$, the implication $M \Downarrow \Rightarrow [\![M]\!] \neq \bot$ is just a reformulation of the convergence lemma for closed terms. The converse implication is a consequence of the soundness of the natural deduction systems with respect to the realizability interpretation, since $[\![M]\!] \neq \bot \Leftrightarrow \vdash M : \gamma \Rightarrow \models M : \gamma \Leftrightarrow M \Downarrow$. ∎

One should note that the adequacy results do not involve the characterization lemma. This means that the canonical domains $\boldsymbol{D}_{\star}$ and $\boldsymbol{D}_{\mathrm{s}}$ provide us with adequate abstract models respectively for the usual (lazy) λ-calculus and the weak call-by-value λ-calculus; see Abramsky and Ong [4] and Moggi [29].

It remains to prove the results mentioned above; that is, weakening, cut and paste, the convergence lemma for closed terms, the characterization lemma, and the extensionality lemma (the deduction theorem). This is done in the next section, but we first give some consequences of completeness and full abstraction.

4.4. *Some Consequences*

A first obvious consequence of full abstraction is that all the previously encountered preorders coincide. Then we can forget about the subscripts, and simply write $M \sqsubseteq N$ and $M \simeq N$. Here we list, without proving them, some properties of this preorder. (These are mostly given for the λ_j^{nv}-calculus. The reader is invited to find the appropriate strict versions.) It is a precongruence, and in particular we have

$$(\mu^*) \quad M \sqsubseteq M' \Rightarrow NM \sqsubseteq NM'$$

$$(\nu^*) \quad M \sqsubseteq M' \Rightarrow MN \sqsubseteq M'N$$

$$(\xi^*) \quad M \sqsubseteq M' \Rightarrow \lambda xM \sqsubseteq \lambda xM'.$$

The equations μ and ξ are semantically valid; therefore they can be used to perform some optimizations. For instance, if we define a left sequential disjunction by $\mathbf{V}_l = \lambda xy.\mathrm{cond}\, x\mathbf{K}y$, then we also have $\mathbf{V}_l \simeq \lambda xy.(x\mathbf{K})\, y$.

We already mentioned that the conversion relation is stronger than the semantical equality, that is, $M \bowtie N \Rightarrow M \simeq N$, since it is a symmetric applicative simulation. Then, for instance, we have

$$M \equiv N \Rightarrow M \simeq N$$

$$\langle \lambda xM, \sigma \rangle\, N \simeq M[N/x]\, \sigma$$

$$N \Downarrow \Rightarrow \langle \lambda^v xM, \sigma \rangle\, N \simeq M[N/x]\, \sigma.$$

As in Scott [42] and Abramsky [3], a weak version of the "functionally principle" η holds; that is,

$$\eta^- : M \sqsubseteq \lambda x(Mx) \qquad (x \text{ not free in } M).$$

Similarly one has $M \sqsubseteq \lambda^v x(Mx)$. Moreover one can check that

$$M \Downarrow \Rightarrow M \simeq \lambda x(Mx) \qquad (x \text{ not free in } M).$$

The ω-rule of the λ-calculus (see Barendregt [5, Definition 4.1.10]) is not valid: for instance, we have $\Omega R \simeq (\lambda x\Omega)\, R$ for all closed terms R, but $\Omega \not\simeq \lambda x\Omega$. However we have, for closed terms,

$$M \sqsubseteq N \Leftrightarrow (M \prec N \text{ and } \forall R \text{ closed. } MR \sqsubseteq NR).$$

Since for closed terms it holds that

$$\models M : \phi \Leftrightarrow \mathbf{M}_\phi \sqsubseteq M,$$

one can see that axiomatization of entailment is *complete*; that is,

$$\phi \leqslant \psi \Leftrightarrow \mathbf{M}_\phi \sqsubseteq \mathbf{M}_\psi \Leftrightarrow \forall M. \models M : \phi \Rightarrow \models M : \psi.$$

The full testing ability is achieved using only *finite* tests, since we have

$$\models M : \phi \Leftrightarrow \mathbf{T}_\phi M \Downarrow .$$

Similarly we could have restricted applicative tests to finite data; that is,

$$A_f ::= [\,] \mid \langle \lambda x A_f, \sigma \rangle \mid \langle \lambda^{\vee} x A_f, \sigma \rangle \mid (A_f \mathbf{M}_\phi).$$

Obviously $(M \parallel N)$ is the least upper bound of M and N,

$$\forall P. M \sqsubseteq P \text{ and } N \sqsubseteq P \quad \Leftrightarrow \quad (M \parallel N) \sqsubseteq P,$$

therefore the usual properties of the join (namely the ones of set-theoretic union) hold:

$$(M \parallel (N \parallel P)) \simeq ((M \parallel N) \parallel P)$$
$$(M \parallel N) \simeq (N \parallel M)$$
$$(M \parallel M) \simeq M$$
$$(M \parallel \Omega) \simeq M.$$

Moreover we have

$$(M \parallel N)\, P \simeq (MP \parallel NP)$$
$$\lambda x(M \parallel N) \simeq (\lambda x M \parallel \lambda x N)$$
$$\lambda^{\vee} x(M \parallel N) \simeq (\lambda^{\vee} x M \parallel \lambda^{\vee} x N).$$

Similarly we have $(MN \parallel MP) \sqsubseteq M(N \parallel P)$, but the converse is false in general. For instance if $t = \gamma \to (\omega \to \gamma)$ and $f = \omega \to (\gamma \to \gamma)$, then we have $\mathbf{T}_{t \wedge f}(\mathbf{K} \parallel \mathbf{F}) \Downarrow$, whereas $\mathbf{T}_{t \wedge f}\mathbf{K} \Uparrow$ and $\mathbf{T}_{t \wedge f}\mathbf{F} \Uparrow$. We indicated in the introduction how to define a parallel disjunction (parallel or):

$$\mathbf{O} =_{\text{def}} \lambda xy.((x\mathbf{T})\, y \parallel ((y\mathbf{T})\, x) = (\mathbf{V}_1 \parallel \mathbf{V}_r).$$

One can check that this combinator satisfies the required properties, namely

$$(\mathbf{O}\Omega)\, \mathbf{T} \simeq \mathbf{T} \simeq (\mathbf{O}\mathbf{T})\, \Omega$$
$$(\mathbf{O}\mathbf{F})\, \mathbf{F} \simeq \mathbf{F}.$$

5. The Logical Systems: Proofs

5.1. *Structural Properties: Weakening and Cut*

Usually in a logical calculus there is no particular "annotation" associated with the formulae: a sequent has the form $\Gamma \vdash \phi$, where $\Gamma = \phi_1, ..., \phi_n$ is a sequence of formulae. In fact this sequence is almost always a multiset, since one usually assumes the *exchange* rule:

$$\frac{\Gamma, \phi, \psi, \Delta \vdash \zeta}{\Gamma, \psi, \phi, \Delta \vdash \zeta}.$$

However, in our systems we can only exchange two items $x : \phi$ and $y : \psi$ in the hypothesis if the variables are distinct. For instance $x : \phi, x : \psi \vdash x : \phi$, but the sequent $x : \psi, x : \phi \vdash x : \phi$, which is not provable unless $\psi \leqslant \phi$, would lead to the wrong conclusion $\vdash \lambda x \lambda x x : \phi \to (\psi \to \phi)$. Similarly, the *contraction* rule

$$\frac{\Gamma, \phi, \phi, \Delta \vdash \zeta}{\Gamma, \phi, \Delta \vdash \zeta}$$

does not really make sense in our systems: obviously we may contract $x : \phi$, $x : \phi$ into $x : \phi$, but it would be absurd to try to contract $x : \phi, y : \phi$ if $x \neq y$. Another rule which is frequently assumed (and, indeed, needed to "type" λ-terms such as **K**) is the *weakening* rule:

$$\frac{\Gamma \vdash \phi}{\Gamma, \Delta \vdash \phi}.$$

To show that this rule holds in our logical systems, we need two preliminary results. Let us first show that we cannot infer more about a variable than assumed:

LEMMA 5.1. $H \vdash x : \phi \Leftrightarrow H(x) \leqslant \phi$.

Proof. The implication $H(x) \leqslant \phi \Rightarrow H \vdash x : \phi$ clearly holds, by virtue of L4 and L3. The converse is easily proved by induction on the inference on the sequent $H \vdash x : \phi$ (which can only be inferred by means of L1, L2, L3, or L4). One uses the general laws E1, E4, E2, and E3.1 for entailement. ∎

Then we remark that proving that a substitution σ satisfies a hypothesis G amounts to showing that each "element" $\sigma(x)$ of the substitution satisfies the corresponding hypothesis:

LEMMA 5.2. $H \vdash \sigma : G \Leftrightarrow \forall x. H \vdash \sigma(x) : G(x)$.

Proof. It is easy to show that $H \vdash \sigma : G \Rightarrow \forall x. H \vdash \sigma(x) : G(x)$, by induction on the inference of $H \vdash \sigma : G$. In the case of $\sigma = \varepsilon$ one uses L4 and L3. Conversely, if $H \vdash \sigma(x) : G(x)$, then one proceeds by induction on σ. In the case of $\sigma = \varepsilon$ one uses the previous lemma. If $\sigma = [M/y]\,\rho$, let $\phi = G(y)$ and $G' = (y : \omega, G)$. Then $H \vdash \rho(x) : G'(x)$ for any variable x; therefore $H \vdash \rho : G'$ by the induction hypothesis. Since $H \vdash M : \phi$ and $(y : \phi, G') = G$ we have $H \vdash \sigma : G$ by S2. ∎

It should be clear that if a sequent $H \vdash \sigma : G$ is used, by means of the rules for the closures, in the inference of a sequent $F \vdash M : \phi$, then for any variable x either $H \vdash \sigma(x) : G(x)$, with a proof shorther than the one of $F \vdash M : \phi$, or $\sigma(x) = x$ and $H(x) \leqslant G(x)$. This fact is used to establish some of the following technical results. In particular, we can now prove our weakening lemma:

LEMMA 5.3 (Weakening). *If* $H \vdash M : \phi$ *and* $G \leqslant^{V} H$ *with* $\mathrm{fv}(M) \subseteq V$ *then* $G \vdash M : \phi$.

Proof. By induction on the inference of the sequent $H \vdash M : \phi$, and by case on the last rule used in this inference:

- This is obvious for L1. For L2 and L3, we simply use the induction hypothesis.
- For L4 we have $M = x$, hence $x \in V$, and $H = (x : \phi, H')$. Then $G \leqslant^{V} H$ implies $G = (x : \psi, G')$ with $\psi \leqslant \phi$. By L4 and L3 we then have $G \vdash x : \phi$.
- For L5.1, $M = \langle \lambda x R, \sigma \rangle$ and $\phi = \psi \to \tau$ with $x : \psi$, $F \vdash_{\star} R : \tau$, and $H \vdash_{\star} \sigma : F$. Let F' be the hypothesis given by

$$F'(z) = \begin{cases} F(z) & \text{if } z \in \mathrm{fv}(R) - \{x\} \\ \omega & \text{otherwise.} \end{cases}$$

Let $W = \mathrm{fv}(R)$. Then $(x : \psi, F') \leqslant^{W}_{\star} (x : \psi, F)$; therefore by the induction hypothesis,

$$x : \psi, F' \vdash_{\star} R : \tau.$$

Let us show that $G \vdash_{\star} \sigma(z) : F'(z)$ for any variable z:

- • This is true if $z \notin \mathrm{fv}(R) - \{x\}$, by L1.
- • If $z \in \mathrm{fv}(R) - \{x\}$ and $H \vdash_{\star} \sigma(z) : F(z)$ with a proof that is shorter than the one of $H \vdash_{\star} M : \phi$, then by induction hypothesis $G \vdash_{\star} \sigma(z) : F(z)$, and $F'(z) = F(z)$.

• • If $z \in \mathrm{fv}(R) - \{x\}$ with $\sigma(z) = z$ and $H(z) \leqslant_\star F(z)$ then $z \in V$, therefore $G(z) \leqslant_\star H(z)$, and $F(z) = F'(z)$, hence $G \vdash_\star \sigma(z) : F'(z)$ be L4 and L3.

Then $G \vdash_\star \sigma : F'$ by Lemma 5.2, hence $G \vdash_\star M : \phi$ by L5.1. The cases of L_s5 and L5.2 are entirely similar.

• For L6 we have $M = (PQ)$ with $H \vdash P : (\psi \to \phi)$ and $H \vdash Q : \psi$. Since $\mathrm{fv}(P) \subseteq \mathrm{fv}(M)$ and $\mathrm{fv}(Q) \subseteq \mathrm{fv}(M)$ we have by the induction hypothesis $G \vdash P : (\psi \to \phi)$ and $G \vdash Q : \psi$, and we use L6 to infer $G \vdash M : \phi$. The argument is similar for L7. ▮

This lemma allows us to restrict the hypothesis to the free variables of the subject term M, assigning ω to any other variable, or to introduce conjunction into the hypotheses: given two hypotheses F and G we define their conjunction $F \wedge G$ by $(F \wedge G)(x) = F(x) \wedge G(x)$ for all x. The following fact is an easy consequence of E3.2, E3.3, and E4:

Remark. $H \leqslant^V F \wedge G \Leftrightarrow H \leqslant^V F$ and $H \leqslant^V G$.

Therefore we have

COROLLARY 5.4. *If* $F \vdash M : \phi$ *or* $G \vdash M : \phi$ *then* $F \wedge G \vdash M : \phi$.

Another structural rule (in the usual logical sense, that is, concerning the structure of the hypotheses) which should hold in any sequent calculus is the *cut* rule, whose usual formulation is

$$\frac{\Gamma \vdash \psi \quad \psi, \Gamma \vdash \phi}{\Gamma \vdash \phi}.$$

In our system, the cut rule results from

PROPOSITION 5.5. $H \vdash M : \phi$ *and* $G \vdash \sigma : H \Rightarrow G \vdash M\sigma : \phi$.

Proof. By induction on the inference of the sequent $H \vdash M : \phi$, and by cases on the last rule used in this inference. The only cases deserving some consideration are L5.1, L_s5 and L5.2. In the case of L5.1 we have $M = \langle \lambda x R, \rho \rangle$ and $\phi = \psi \to \tau$ with $x : \psi$, $F \vdash_\star R : \tau$, and $H \vdash_\star \rho : F$. Let us show that $G \vdash_\star (\rho \circ \sigma)(z) : F(z)$ for any variable z: if $H \vdash_\star \rho(z) : F(z)$ with a proof that is shorter than the one of $H \vdash_\star M : \phi$, then by the induction hypothesis $G \vdash_\star (\rho \circ \sigma)(z) : F(z)$ since $(\rho \circ \sigma)(z) = (\rho(z))\,\sigma$ (Lemma 2.1). Otherwise $\rho(z) = z$ and $H(z) \leqslant_\star F(z)$, in which case $(\rho \circ \sigma)(z) = \sigma(z)$. Since $G \vdash_\star \sigma(z) : H(z)$ (Lemma 5.2) we have $G(z) \vdash_\star \sigma(z) : F(z)$ by L3. Then $G \vdash_\star (\rho \circ \sigma) : F$ by Lemma 5.2, and $G \vdash_\star M\sigma : \phi$ by L5.1 since $M\sigma = \langle \lambda x R, \rho \circ \sigma \rangle$. The cases of L_s5 and L5.2 are entirely similar. ▮

COROLLARY 5.6 (Cut). *$H \vdash N : \psi$ and $x : \psi, H \vdash M : \phi \Rightarrow H \vdash M([N/x]\,\varepsilon) : \phi$.*

Proof. By S1 we have $H \vdash \varepsilon : H$, therefore $H \vdash [N/x]\,\varepsilon : (x : \psi, H)$ by S2. ∎

5.2. *Extensionality, Paste, and Reduction*

In this section we first prove a refined version of the deduction theorem. The usual statement of the deduction theorem, in logical systems without "annotations," is

$$\Gamma \vdash \phi \to \psi \quad \Leftrightarrow \quad \phi, \Gamma \vdash \psi.$$

Assuming that in the strict case $\phi \to \psi$ is a well-formed formula, that is, $\phi \in \Gamma$, it is very easy to prove that this holds in our logical systems: in the "$\Leftarrow$" direction, this is just L5.1 or L_s5. For the converse, if $H \vdash M : \phi \to \psi$ and x is a variable not in $\mathrm{fv}(M)$, then $x : \phi, H \vdash M : \phi \to \psi$ by weakening, and $x : \phi, H \vdash x : \phi$ by L4, therefore $x : \phi, H \vdash Mx : \psi$ by L6 (modus ponens). However, if we abstract once more, we get the term λxMx or $\lambda^{\vee} xMx$, which is usually more informative than M (for instance if $M = \Omega$ or $M = \Omega_s$). As indicated above, we need a more refined result. Recall that γ denotes the formula $\omega \to \omega$ in the $\Phi_\star$-logic. Then our sharpened deduction theorem is

EXTENSIONALITY LEMMA 5.7 (The Deduction Theorem). *$H \vdash M : \phi \to \psi$ if and only if $H \vdash M : \gamma$ and $x : \phi, H \vdash Mx : \psi$ for some $x \notin \mathrm{fv}(M)$ (assuming that $\phi \in \Gamma$ in the strict case).*

Proof. If $H \vdash_\star M : \phi \to \psi$ then we have $H \vdash_\star M : \omega \to \omega$ by L3, since by E1, $E_\star 7$, and $E_\star 5$,

$$\phi \to \psi \leqslant_\star \phi \to \omega \leqslant_\star \omega \to \omega.$$

Similarly in the strict case $H \vdash_s M : \gamma$ since $\phi \to \psi \leqslant_s \gamma$ by $E_s 5.1$. Moreover if $x \notin \mathrm{fv}(M)$ we have by weakening $x : \phi, H \vdash M : \phi \to \psi$, and $x : \phi, H \vdash x : \phi$ by L4. Then by L6 we get $x : \phi, H \vdash Mx : \psi$.

Conversely, assume that $H \vdash M : \gamma$ and $x : \phi, H \vdash Mx : \psi$ with $x \notin \mathrm{fv}(M)$, and $\phi \in \Gamma$ in the strict case. We prove that $H \vdash M : \phi \to \psi$ by induction on the inference of $x : \phi, H \vdash Mx : \psi$:

- If this sequent is proved using L1 then $\psi = \omega$, and $H \vdash M : (\phi \to \psi\omega)$ by L3 since $H \vdash M : \gamma$ and $\gamma \leqslant \phi \to \omega$ (by E1 and $E_\star 7$, or $E_s 5.2$, $E_s 5.1$, and $E_s 7$).

• By L2, we have $\psi = \psi_0 \wedge \psi_1$, and by the induction hypothesis $H \vdash M : \phi \to \psi_i$. Then by L2 $H \vdash M : (\phi \to \psi_0) \wedge (\phi \to \psi_1)$; hence by L3 and $E_\star 6$ (or $E_s 6$) we get $H \vdash M : \phi \to (\psi_0 \wedge \psi_1)$.

• By L3, $x : \phi$, $H \vdash Mx : \psi'$ for some ψ' such that $\psi' \leqslant \psi$. Then by the induction hypothesis $H \vdash M : \phi \to \psi'$, hence $H \vdash M : \phi \to \psi$ by L3, since $\phi \to \psi' \leqslant \phi \to \psi$ (by $E_\star 7$ or $E_s 7$).

• The only remaining case is L6. Then for some ϕ' we have $x : \phi$, $H \vdash M : \phi' \to \psi$, and $x : \phi$, $H \vdash x : \phi'$. By Lemma 5.1, $\phi \leqslant \phi'$; therefore by L3, $x : \phi$, $H \vdash M : \phi \to \psi$ since $\phi' \to \psi \leqslant \phi \to \psi$ (by $E_\star 7$ or $E_s 7$). Since $x \notin \mathrm{fv}(M)$ we have $H \vdash M : \phi \to \psi$ by weakening. ∎

An important property of Coppo's typing systems is that if a term M reduces N and N satisfies ϕ under the hypothesis H, then the same holds for M (cf. [10, 21]). Since we defined reduction only for closed terms, we show here that

$$M \rhd N \text{ and } \vdash N : \phi \quad \Rightarrow \quad \vdash M : \phi.$$

In other words, reduction is decreasing with respect to "typability," that is, with respect to the semantical preorder. To establish this property in the case of the β-rules, we need to know how to "type" $R[Q/x]\,\rho$. This is expressed in the paste property, proved below. Let us first see some auxiliary properties. One can remark that the rules L1, L2, and L3 can be extended to substitutions:

LEMMA 5.8. (i) *If $G(x) = \omega$ for any x then $H \vdash \sigma : G$ for any H and σ.*

(ii) *If $H \vdash \sigma : G_0$ and $H \vdash \sigma : G_1$ then $H \vdash \sigma : G_0 \wedge G_1$.*

(iii) *If $H \vdash \sigma : G$ and $G \leqslant F$ then $H \vdash \sigma : F$.*

Proof. Trivial. ∎

Similarly, the weakening lemma holds for the sequents concerning substitutions; that is,

LEMMA 5.9. *$H \vdash \sigma : G$ and $F \leqslant H \Rightarrow F \vdash \sigma : G$.*

Now we can prove the paste property:

PROPOSITION 5.10 (Paste). *$H \vdash M\sigma : \phi \Rightarrow \exists G.\, H \vdash \sigma : G$ and $G \vdash M : \phi$.*

Proof. We show simultaneously

$$H \vdash M\sigma : \phi \quad \Rightarrow \quad \exists G.\, H \vdash \sigma : G \text{ and } G \vdash M : \phi$$

and

$$H \vdash \rho \circ \sigma : F \quad \Rightarrow \quad \exists G.\, H \vdash \sigma : G \text{ and } G \vdash \rho : F$$

by structural induction on M and ρ, and then by induction on the proofs of the sequents. The sequent $H \vdash M\sigma : \phi$ can only be proved using one of the "filter rules" L1, L2, or L3, or using a specific rule depending on the structure of M. In fact we can "factorize" the cases of L1, L2, and L3, regardless of the shape of M:

• If $H \vdash M\sigma : \phi$ is proved using L1, then $\phi = \omega$ and we can let $G(x) = \omega$ for any x. Then $H \vdash \sigma : G$ by Lemma 5.8, and $G \vdash M : \phi$ by L1.

• If $H \vdash M\sigma : \phi$ is proved using L2 then $\phi = \phi_0 \wedge \phi_1$ with $H \vdash M\sigma : \phi_i$. Then by the induction hypothesis (on the inference of the sequents) there exists G_i such that $H \vdash \sigma : G_i$ and $G_i \vdash M : \phi_i$. Let $G = G_0 \wedge G_1$; then by Lemma 5.8 above $H \vdash \sigma : G$, and $G \vdash M : \phi_i$ by weakening, therefore $G \vdash M : \phi$ by L2.

• If $H \vdash M\sigma : \phi$ is proved using L3, there exists ψ such that $\psi \leqslant \phi$ and $H \vdash M\sigma : \psi$. By the induction hypothesis (on the proof of the sequents) there exists G such that $H \vdash \sigma : G$ and $G \vdash M : \psi$, therefore $G \vdash M : \phi$ by L3.

Now we examine the various possibilities for M:

• If is a variable, say x, then we let $G = x : \phi$. We have $H \vdash \sigma(y) : G(y)$ for any variable y since $\sigma(x) = M\sigma$, therefore $H \vdash \sigma : G$ by Lemma 5.2, and $G \vdash x : \phi$ by L4.

• If $M = \langle \lambda x R, \rho \rangle$, then $M\sigma = \langle \lambda x R, \rho \circ \sigma \rangle$, and we may assume that $H \vdash M\sigma : \phi$ is proved using L5.1; that is, $\phi = \psi \to \tau$ with $x : \psi$, $F \vdash R : \tau$ and $H \vdash \rho \circ \sigma : F$. By the induction hypothesis (recall that $\rho \propto M$) there exists G such that $H \vdash \sigma : G$ and $G \vdash \rho : F$. Then $G \vdash M : \phi$ by L5.1. The proof is the same for $M = \langle \lambda^{\nu} x R, \rho \rangle$ (with $H \vdash M : \phi$ inferred by means of L5.2 or $L_s 5$).

• If $M = M_0 M_1$, we have $M\sigma = M_0\sigma M_1\sigma$, and we may assume that $H \vdash M\sigma : \phi$ is proved using L6; that is, $H \vdash M_0\sigma : \psi \to \phi$ and $H \vdash M_1\sigma : \psi$. By the induction hypothesis there exist G_0 and G_1 such that $H \vdash \sigma : G_i$ and $G_0 \vdash M_0 : \psi \to \phi$ and $G_1 \vdash M_1 : \psi$. Let $G = G_0 \wedge G_1$; then $H \vdash \sigma : G$ by Lemma 5.8 above, and $G \vdash M_0 : \psi \to \phi$ and $G \vdash M_1 : \psi$ by weakening, therefore $G \vdash M : \phi$ by L6.

• If $M = (M_0 \parallel M_1)$ then $M\sigma = (M_0\sigma \parallel M_1\sigma)$, and we may assume that $H \vdash M\sigma : \phi$ is proved by means of L7, that is, $\phi = \phi_0 \wedge \phi_1$ with $H \vdash M_i\sigma : \phi_i$. By the induction hypothesis there exists G_i such that $H \vdash \sigma : G_i$ and $G_i \vdash M_i : \phi_i$. Let $G = G_0 \wedge G_1$; then by Lemma 5.8 above $H \vdash \sigma : G$ and $G \vdash M_i : \phi_i$ be weakening, therefore $G \vdash M : \phi$ by L7.

Finally we examine the two possible cases for ρ:

• If $\rho = \varepsilon$ then $\rho \circ \sigma = \sigma$. We let $G = F$; then $H \vdash \sigma : G$ and $G \vdash \rho : F$ by S1.

• If $\rho = [M/x]\,\rho'$ then $\rho \circ \sigma = [M\sigma/x](\rho' \circ \sigma)$, and $H \vdash \rho \circ \sigma : F$ can only be proved using S2; that is, $F = (x : \phi, F')$ with $H \vdash M\sigma : \phi$ and $H \vdash \rho' \circ \sigma : F'$. Then by the induction hypothesis (recall that $M \propto \rho$ and $\rho' \propto \rho$) there exist G' and G'' such that $H \vdash \sigma : G'$ and $G' \vdash M : \phi$, and $H \vdash \sigma : G''$ and $G'' \vdash \rho' : F'$. Let $G = G' \wedge G''$. Then by Lemma 5.8 above, $H \vdash \sigma : G$, and $G \vdash M : \phi$ by weakening, and $G \vdash \rho' : F'$ by Lemma 5.9. Therefore $G \vdash \rho : F$ by S2. ▮

To show that reduction is semantically decreasing, we need some preliminary results. We let the reader check that the following holds:

Remark 5.11. $\forall \phi \in \Phi_s\, \exists \zeta \in \Gamma.\, \phi \wedge \gamma \sim_s \zeta$.

Let us show that any partial normal form satisfies the convergence formula:

LEMMA 5.12. *If K is a partial normal form then $\vdash K : \gamma$.*

Proof. By induction on the definition of the notion of partial normal form.

• If K is a closure $\langle \lambda x R, \rho \rangle$, let G be the hypothesis given by $G(z) = \omega$ for any variable z. Then $x : \omega$, $G \vdash_\star R : \omega$ by L1, and $\vdash_\star \rho : G$ (by S1, or by a repeated use of L1 and S2 if $\rho \neq \varepsilon$). Then $\vdash_\star K : \gamma$ by L5.1.

• If $K = \langle \lambda^\vee x R, \rho \rangle$, let G be the hypothesis given by $G(z) = \omega$ for any variable z. Then $x : \gamma$, $G \vdash_\star R : \omega$ by L1, and $\vdash_\star \rho : G$. Then $\vdash_\star K : \gamma \to \omega$ by L5.2, therefore $\vdash_\star K : \gamma$ by $E_\star 5$ and L3. Similarly $\vdash_s K : \gamma \to \omega$ by $L_s 5$, hence $\vdash_s K : \gamma$ by $E_s 5.1$ and L3.

• If K is $(M \parallel N)$ where M or N is a partial normal form, then we use the induction hypothesis and the fact that $\vdash P_i : \phi \Rightarrow \vdash (P_0 \parallel P_1) : \phi$ (see Section 3.3.2). ▮

PROPOSITION 5.13 (Reduction is Decreasing). *$M \rhd N$ and $\vdash N : \phi \Rightarrow \vdash M : \phi$.*

Proof. By induction on the definition of $M \rhd N$, and then on the inference of $\vdash N : \phi$. As in the proof of the paste property, it is easy to see that one can "factorize" the cases where this sequent is proved by means of L1, L2, or L3. This is left to the reader. In the rest of the proof we do not take these cases into consideration.

• If $M = \langle \lambda x R, \sigma \rangle$ and $N = \langle \lambda x R, \rho \rangle$ with $\sigma(z) \rhd \rho(z)$ for $z \in \mathrm{fv}(R) - \{x\}$, we may assume that $\vdash N : \phi$ is proved by means of L5.1;

that is, $\phi = \psi \to \tau$ with $x : \psi$, $H \vdash R : \tau$, and $\vdash \rho : H$. Let G be the hypothesis given by

$$G(z) = \begin{cases} H(z) & \text{if } z \in \mathrm{fv}(R) - \{x\} \\ \omega & \text{otherwise.} \end{cases}$$

Then $x : \psi$, $G \vdash R : \tau$ by weakening and $\vdash \rho : G$ by Lemma 5.8, since $H \leqslant G$. By Lemma 5.2 we have $\vdash \rho(z) : G(z)$ for any z; therefore by the induction hypothesis (since the proof of $\sigma(z) \rhd \rho(z)$ is shorter than that of $M \rhd N$) $\vdash \sigma(z) : G(z)$ for $z \in \mathrm{fv}(R) - \{x\}$, and $\vdash \sigma(z) : G(z)$ for $z \notin \mathrm{fv}(R) - \{x\}$ by L1. Then $\vdash \sigma : G$ by Lemma 5.2, and $\vdash M : \phi$ by L5.1.

• The proof is the same for $M = \langle \lambda^{\mathrm{v}} xR, \sigma \rangle$ and $N = \langle \lambda^{\mathrm{v}}, \rho \rangle$ with $\sigma(z) \rhd \rho(z)$ for $z \in \mathrm{fv}(R) - \{x\}$.

• If $M = M_0 M_1$ and $N = N_0 N_1$ with $M_i \rhd N_i$, we may assume that $\vdash N : \phi$ is proved by L6; that is, $\vdash N_0 : \psi \to \phi$ and $\vdash N_1 : \psi$. Then we use the induction hypothesis and L6 to show that $\vdash M : \phi$. The argument is similar if $M = (M_0 \parallel M_1)$ and $N = (N_0 \parallel N_1)$ with $M_i \rhd N_i$, and $\vdash N : \phi$ is proved using L7.

• If $M = \langle \lambda xR, \sigma \rangle P$ and $N = R[P/x]\,\sigma$ then by paste there exists H such that $\vdash [P/x]\,\sigma : H$ and $H \vdash R : \phi$. The first sequent can only be inferred using S2; that is, $H = (x : \psi, G)$ for some ψ and G such that $\vdash P : \psi$ and $\vdash \sigma : G$. Then $x : \psi$, $G \vdash R : \phi$, therefore $\vdash \langle \lambda xR, \sigma \rangle : \phi$ by L5.1, and $\vdash M : \phi$ by L6 since $\vdash P : \psi$.

• If $M = \langle \lambda^{\mathrm{v}} xR, \sigma \rangle K$ and $N = R[K/x]\,\sigma$, where K is a partial normal form, then we may assume that $\vdash N : \phi$ is proved by means of L5.2 (in the $\lambda_{\mathrm{j}}^{\mathrm{nv}}$-calculus) or by means of $\mathrm{L_s}5$ (in the $\lambda_{\mathrm{j}}^{\mathrm{v}}$-calculus). In any case, by paste there exists H such that $\vdash [K/x]\,\sigma : H$ and $H \vdash R : \phi$. Then there exist ψ and G such that $H = (x : \psi, G)$, and $\vdash [K/x]\,\sigma : H$ is inferred, by S2, from $\vdash K : \psi$ and $\vdash \sigma : G$. By Lemma 5.12 we have $\vdash K : \gamma$. Then in the $\lambda_{\mathrm{j}}^{\mathrm{nv}}$-calculus we have $x : \psi \wedge \gamma$, $G \vdash R : \phi$ by weakening, therefore $\vdash \langle \lambda^{\mathrm{v}} xR, \sigma \rangle : \psi \wedge \gamma \to \phi$ since $\psi \wedge \gamma \leqslant_{\star} \omega \to \omega$. Since $\vdash K : \psi \wedge \gamma$ by L2, we get $\vdash M : \phi$ by L6. The proof is similar in the $\lambda_{\mathrm{j}}^{\mathrm{nv}}$-calculus since we know by Remark 5.11 that there exists $\zeta \in \Gamma$ such that $\zeta \sim_{\mathrm{s}} \psi \wedge \gamma$; therefore $x : \zeta$, $G \vdash R : \phi$, and $\vdash K : \zeta$.

• The only remaining case is $M = (M_0 \parallel M_1)\,P$ and $N = (M_0 P \parallel M_1 P)$. We may assume that $\vdash N : \phi$ is proved using L7; that is, $\phi = \phi_0 \wedge \phi_1$ with $\vdash M_i P : \phi_i$. These sequents can be inferred by means of L1, L2, L3, or L6. By symmetry, we can restrict our investigations to the following cases:

•• $\vdash M_0 P : \phi_0$ and $\vdash M_1 P : \phi_1$ are both proved by L1; i.e., $\phi_i = \omega$. Then $\phi \sim \omega$, therefore $\vdash M : \phi$ by L1 and L3.

• • $\vdash M_0 P : \phi_0$ is proved using L2; that is, $\phi_0 = \psi_0 \wedge \psi_1$ with $\vdash M_0 P : \psi_i$. Then $\vdash N : \psi_i \wedge \phi_1$ (by L7) with a proof shorter than that of $\vdash N : \phi$, therefore by the induction hypothesis $\vdash M : \psi_i \wedge \phi_1$, hence $\vdash M : \phi$ by L2 and L3 since $(\psi_0 \wedge \phi_1) \wedge (\psi_1 \wedge \phi_1) \sim \phi$.

• • If $\vdash M_0 P : \phi_0$ is proved by means of L3, then there exist $\psi \leqslant \phi_0$ such that $\vdash M_0 P : \psi$. Then $\vdash N : \psi \wedge \phi_1$ (by L7) with a proof shorter than that of $\vdash N : \phi$, therefore by the induction hypothesis $\vdash N : \psi \wedge \phi_1$, hence $\vdash M : \phi$ by L3 since $\psi \wedge \phi_1 \leqslant \phi$.

• • If $\vdash M_0 P : \phi_0$ by L6, we have $\vdash M_0 : \psi_0 \to \phi_0$ and $\vdash P : \psi_0$ for some ψ_0. Note that $\vdash (M_0 \parallel M_1) : \psi_0 \to \phi_0$ (see Section 3.3.2), therefore $\vdash M : \phi_0$ by L6. Here we only have two cases to take into consideration: if $\vdash M_1 P : \phi_1$ by L1, that is $\phi_1 = \omega$, then $\vdash M : \phi$ by L3 since $\phi = \phi_0 \wedge \omega \sim \phi_0$. Otherwise we may assume that $\vdash M_1 P : \phi_1$ inferred using L6, that is $\vdash M_1 : \psi_1 \to \phi_1$ and $\vdash P : \psi_1$ for some ψ_1. As above we have $\vdash (M_0 \parallel M_1) : \psi_1 \to \phi_1$, therefore $\vdash M : \phi_1$ by L6, and $\vdash M : \phi$ by L2. ∎

We can now prove the convergence lemma for closed terms, as an easy consequence of the previous proposition:

COROLLARY 5.14. *If M is a closed term then $M \Downarrow \Rightarrow \vdash M : \gamma$.*

Proof. If $M \Downarrow$ then by definition there exists a closed partial normal form K such that $M \Downarrow K$. Then by Lemma 2.10 we have $M \rhd^* K$, and $\vdash K : \gamma$ by Lemma 5.12. It should be obvious, from the previous proposition, that $N \rhd^* R$ and $\vdash R : \phi \Rightarrow \vdash N : \phi$, therefore $\vdash M : \gamma$. ∎

5.3. *Characteristic Terms*

To establish our last result, that is, the characterization lemma, it will be convenient to use a restricted natural deduction system, which although incomplete, allows us to prove the "essential" properties (i.e, formulae) of the terms. This restricted logical system is quite close to the original system of Coppo [10, 11]. The sequents in this system are denoted $H \Vdash M : \phi$, though the restriction actually concerns the formulae. To see what are the special formulae we use, let us define inductively the *decomposition* relation $\phi \rhd \psi$ between formulae of $\Phi_\star$ or Φ_s, as follows:

$$\text{(i)} \quad \omega \rhd \omega \qquad \text{and} \qquad \gamma \rhd \gamma$$

$$\text{(ii)} \quad \phi \rhd \pi \Rightarrow \begin{cases} \phi \wedge \psi \rhd \pi \\ \psi \wedge \phi \rhd \pi \\ \psi \to \phi \rhd \psi \to \pi. \end{cases}$$

Remark 5.15. $\phi \rhd \psi \Rightarrow \phi \leqslant \psi$.

DEFINITION (Irreducible Formulae). A formula ϕ is *irreducible* if $\phi \rhd \pi \Rightarrow \pi = \phi$.

We denote by $\Pi_\star$ and Π_s the sets of irreducible formulae. Clearly these sets are given by the grammars

$$(\Pi_\star) \quad \pi ::= \omega \mid (\phi \to \pi),$$

where ϕ is any formula of $\Phi_\star$, and

$$(\Pi_s) \quad \pi ::= \omega \mid \gamma \mid (\zeta \to \pi),$$

where ζ is any formula of Γ. In what follows we mostly use the symbol Π, without subscript, for the set of irreducible formulae. An obvious observation is that if $\phi \rhd \psi$ then ψ is an irreducible formula. Let us see that any formula is equivalent to a conjection of irreducible ones. Recall that

$$\phi \sim \psi \quad \Leftrightarrow \quad \phi \leqslant \psi \text{ and } \psi \leqslant \phi.$$

LEMMA 5.16. *For any ϕ the set $\phi^\star = \{\pi \mid \phi \rhd \pi\}$ is non-empty and finite. Moreover $\phi \sim \bigwedge \{\pi \mid \phi \rhd \pi\}$.*

Proof (Sketch). By induction on the definition of $\phi \rhd \pi$, trivial. Note that $(\phi \wedge \psi)^\star = \phi^\star \cup \psi^\star$. For the case $\psi \to \phi$ one uses the equivalence $(\psi \to \bigwedge_{1 \leqslant i \leqslant n} \phi_i) \sim \bigwedge_{1 \leqslant i \leqslant n} (\psi \to \phi_i)$. ∎

A consequence of this fact is that any irreducible formula (distinct from ω and γ) can be written, up to $\sim$, as $\pi_1 \wedge \cdots \wedge \pi_k \to \pi$, where π and the π_i's are irreducible. We could also use Coppo's notation (cf. [11]) for irreducible formulae, that is, $[\pi_1, \ldots, \pi_k]\,\pi$.

Now we characterize the entailment relation $\phi \leqslant \psi$ by means of irreducible formulae (cf. Barendregt *et al.* [6], Hindley [20], Abramsky [3]). Let $\ll_\star$ and $\ll_s$ be the least relations on $\Pi_\star$ and Π_s satisfying

$$\text{F1:} \quad \pi \ll \omega \qquad \text{F2:} \quad \frac{\pi_0 \ll \pi,\ \pi \ll \pi_1}{\pi_0 \ll \pi_1} \qquad \text{F3:} \quad \pi \ll \pi$$

and, for the first one,

$$\text{F}_\star\text{4:} \quad (\phi \to \omega) \ll_\star (\omega \to \omega) \qquad \text{F}_\star\text{5:} \quad \frac{\phi_0 \geqslant_\star \phi_1,\ \pi_0 \ll_\star \pi_1}{(\phi_0 \to \pi_0) \ll_\star (\phi_1 \to \pi_1)},$$

while in the strict case

$$\text{F}_s\text{4.1:} \quad (\zeta \to \pi) \ll_s \gamma \qquad \text{F}_s\text{4.2:} \quad \gamma \ll_s (\gamma \to \omega)$$

$$\text{F}_s\text{5:} \quad \frac{\zeta_0 \geqslant_s \zeta_1,\ \pi_0 \ll_s \pi_1}{(\zeta_0 - \pi_0) \ll_s (\zeta_1 \to \pi_1)}.$$

LEMMA 5.17. $\phi \leqslant \psi \Leftrightarrow \forall \psi'(\psi \rhd \psi' \Rightarrow \exists \phi'(\phi \rhd \phi' \text{ and } \phi' \ll \psi'))$.

Proof (Sketch). The "$\Leftarrow$" part is clear from the previous lemma. The converse implication is easily established by induction on the definition of $\phi \leqslant \psi$. For the cases of $E_s 5.1$ and $E_s 5.2$ one remarks that if $\zeta \in \Gamma$ and $\zeta \rhd \pi$ then $\pi = \gamma$ or $\pi = (\zeta' \to \pi')$ with $\zeta' \in \Gamma$ and $\pi' \in \Pi_s$. ▮

We could have written this property as

$$\phi \leqslant \psi \Leftrightarrow \forall \psi' \in \psi^\star \, \exists \phi' \in \phi^\star . \phi' \ll \psi'.$$

Let us see some consequences of this characterization of the entailment relation. For instance, the reader could show that the irreducible formulae represent the *compact prime* elements of the canonical domains:

LEMMA. *A filter F compact prime if and only if there exists an irreducible formula* $\pi \in \Pi$ *such that* $F = \pi\uparrow$.

Then the previous lemma is just a reformulation of the fact that the canonical domains are also powerdomains; see Section 4.1.1. We can also characterize the formulae generating the least element of the domain:

DEFINITION (Trivial Formulae). A formula ϕ is trivial if $\phi \sim \omega$.

Clearly ϕ is trivial if and only if $\omega \leqslant \phi$. Then $\phi \wedge \psi$ is trivial if and only if both ϕ and ψ are trivial.

LEMMA 5.18. (i) $\phi \sim \omega \Leftrightarrow \phi^\star = \{\omega\}$.

(ii) $\phi \not\sim \omega \Leftrightarrow \phi \leqslant \gamma$ *and, in the strict case,* $(\zeta \to \phi) \sim_s \gamma \Leftrightarrow \phi \sim_s \omega$.

Proof. (i) If $\phi \sim \omega$ then $\omega \leqslant \phi$ and by the previous lemma there exists $\pi \in \phi^\star$ such that $\omega \ll \pi$. It is easy to see, by induction on the definition of $\ll$, that $\omega \ll \pi \Leftrightarrow \pi = \omega$. Therefore $\phi^\star = \{\omega\}$.

(ii) Since $\gamma^\star = \{\gamma\}$, the formula γ is non-trivial, by the previous point, and clearly $\phi \leqslant \psi \not\sim \omega \Rightarrow \phi \not\sim \omega$. Conversely if $\phi \not\sim \omega$ then by the previous lemma there exists $\pi \in \phi^\star$ such that $\omega \not\ll \pi$. It is obvious that this implies that $\pi = \phi \to \tau$, since π is irreducible, or in the strict case $\pi = \gamma$ or $\pi = \zeta \to \tau$, hence $\pi \leqslant \gamma$ by E1, $E_\star 7$, and $E_\star 5$, or by $E_s 5.1$. Then $\phi \leqslant \gamma$ since $\phi \leqslant \pi$ (Remark 5.15). The last point is left to the reader. ▮

Now we introduce the restricted logical systems, which essentially consists in disallowing L2 and L3, and in proving only irreducible assertions about the terms. The two restricted systems share the following rules:

$$\text{T1: } H \Vdash M : \omega \qquad \text{T2: } \frac{}{x : \phi,\ H \Vdash x : \pi}\ \phi \rhd \pi$$

$$\text{T4: } \frac{H \Vdash M : (\phi \to \psi),\ H \Vdash N : \phi_i \quad (1 \leqslant i \leqslant k)}{H \Vdash (MN) : \psi}\ \phi_1 \wedge \cdots \wedge \phi_k \leqslant \phi$$

$$\text{T5.1: } \frac{H \Vdash M : \phi}{H \Vdash (M \parallel N) : \phi} \qquad \text{T5.2: } \frac{H \Vdash N : \psi}{H \Vdash (M \parallel N) : \psi}.$$

The rules for closures are the same as L5.1, L5.2, and L_s5:

$$\text{T3.1: } \frac{x : \psi,\ G \Vdash_\star M : \phi,\ H \vdash_\star \sigma : G}{H \Vdash_\star \langle \lambda x M, \sigma \rangle : (\psi \to \phi)}$$

$$\text{T3.2: } \frac{x : \psi,\ G \Vdash_\star M : \phi,\ H \vdash_\star \sigma : G}{H \Vdash_\star \langle \lambda^{\vee} x M, \sigma \rangle : (\psi \to \phi)}\ \psi \leqslant_\star (\omega \to \omega)$$

$$\text{T}_s\text{3: } \frac{x : \zeta,\ G \Vdash_s M : \phi,\ H \vdash_s \sigma : G}{H \Vdash_s \langle \lambda^{\vee} x M, \sigma \rangle : (\zeta \to \phi)}.$$

It is easily checked, by induction on the inference of the sequents, that

Remark. If $H \Vdash M : \phi$ then ϕ is an irreducible formula.

Remark 5.19. $x : \phi,\ H \Vdash x : \psi \Rightarrow \phi \leqslant \psi$.

Now we relate the logical systems, showing first that the sequents provable in the restricted systems are also provable in the full logical systems, and that the restricted assertions that can be made about a term generate the filter of all possible properties of this term.

PROPOSITION 5.20. (i) $H \Vdash M : \phi \Rightarrow H \vdash M : \phi$.

(ii) $H \vdash M : \phi \Rightarrow \exists \pi_1, \ldots, \pi_n \in \Pi.\, \pi_1 \wedge \cdots \wedge \pi_n \leqslant \phi$ *and* $\forall i.\, H \Vdash M : \pi_i$.

Proof. One easily proves the first point by induction on the inference of the sequent $H \Vdash M : \phi$, using Remark 5.19 above. Let us just check the case of T4: we have $H \Vdash P : \phi \to \psi$ and $H \Vdash Q : \phi_i$ with $\phi_1 \wedge \cdots \wedge \phi_k \leqslant \phi$. Then by the induction hypothesis $H \vdash P : \phi \to \psi$ and $H \vdash Q : \phi_i$; therefore $H \vdash Q : \phi$ by L2 and L3; hence $H \vdash (PQ) : \psi$ by L6.

Now we prove the second point by induction on the inference of the sequent $H \vdash M : \phi$, and by a case analysis on the last rule used in the proof. This is trivial for L1, L2, L3 (using the induction hypothesis), and L4 (using Lemma 5.16).

• For L5.1 we have $\phi = \xi \to \tau$ and $M = \langle \lambda x R, \sigma \rangle$, with $x : \xi$, $G \vdash_\star R : \tau$ and $H \vdash_\star \sigma : G$. Then by the induction hypothesis there exist $\tau_1, \ldots, \tau_n$ such that $\tau_1 \wedge \cdots \wedge \tau_n \leqslant \tau$ and $x : \xi,\ G \Vdash_\star R : \tau_i$. We let

$\pi_i = \xi \to \tau_i$; these are irreducible formulae, since the τ_i are irreducible. By T3.1 we have $H \Vdash_\star M : \pi_i$ for all i, and

$$\pi_1 \wedge \cdots \wedge \pi_n \sim \left(\xi \to \bigwedge_{1 \leqslant i \leqslant n} \tau_i\right) \leqslant \xi \to \tau.$$

The proof is the same for L5.2 and L_s5.

• For L6 we have $M = (PQ)$ with $H \vdash P : \psi \to \phi$ and $H \vdash Q : \psi$. We may assume that $\phi \neq \omega$, since $H \Vdash M : \omega$ can be proved by means of T1. Then if we let $\{\phi_1, ..., \phi_n\} = \{\phi' \mid \phi' \neq \omega \text{ and } \phi \rhd \phi'\}$ we have $\phi \sim \phi_1 \wedge \cdots \wedge \phi_n$. By the induction hypothesis there exist $\tau_1, ..., \tau_k$ such that $\tau_1 \wedge \cdots \wedge \tau_k \leqslant \psi \to \phi$ and $H \Vdash P : \tau_i$. By Lemma 5.17 for any j $(1 \leqslant j \leqslant n)$ there exists i_j such that $\tau_{i_j} \ll \psi \to \phi_j$. Since this can only be proved by means of F2, F3, and $F_\star5$ or F_s5, we clearly have $\tau_{i_j} = \psi_j - \pi_j$ with $\psi_j \geqslant \psi$ and $\pi_j \ll \phi_j$, therefore $\pi_1 \wedge \cdots \wedge \pi_n \leqslant \phi$. By the induction hypothesis on $H \vdash Q : \psi$ there exist $\xi_1, ..., \xi_m$ such that $\xi_1 \wedge \cdots \wedge \xi_m \leqslant \psi$ and $H \Vdash Q : \xi_h$. Then $\xi_1 \wedge \cdots \wedge \xi_m \leqslant \psi_j$ for any j; therefore $H \Vdash M : \pi_j$ for any j, by T4.

The case of L7 is trivial (using the induction hypothesis). ∎

Let us now introduce the characteristic terms and show that they satisfy their specification. We define for each formula $\phi \in \Phi_\star$, and respectively in the strict case for each $\phi \in \Omega_s$ and $\zeta \in \Gamma$, a pair of closed λ_j^{nv}-terms (resp. λ_j^{v}-terms) $\mathbf{M}_\phi$ and $\mathbf{T}_\phi$ (resp. $\mathbf{M}_\phi^s$ and $\mathbf{T}_\zeta^s$) such that

(i) $\vdash_\star \mathbf{M}_\phi : \phi$

(ii) $\vdash_\star \mathbf{T}_\phi : \phi \to (\psi \to \psi)$ for any $\psi \in \Phi_\star$.

and

(iii) $\vdash_s \mathbf{M}_\phi^s : \phi$

(iv) $\vdash_s \mathbf{T}_\zeta^s : \zeta \to (\psi \to \psi)$ for any $\psi \in \Gamma$.

We define these terms inductively, checking the required properties:

$$\mathbf{M}_\omega = \Omega$$

$$\mathbf{M}_\omega^s = \Omega_s$$

$$\mathbf{T}_\omega = \mathbf{F} = \lambda y \lambda x x.$$

Clearly $\vdash \mathbf{M}_\omega : \omega$ by L1, and similarly for $\mathbf{M}_\omega^s$. To show that $\vdash \mathbf{F} : \omega \to (\psi \to \psi)$ is an easy exercise.

$$\mathbf{M}_\gamma^s = \lambda^v x \Omega_s$$

$$\mathbf{T}_\gamma^s = \lambda^v y \lambda^v x x.$$

We have $x:\gamma \vdash \Omega_s:\omega$ by L1, therefore $\vdash \mathbf{M}^s_\gamma : \gamma \to \omega$ by L_s5 (recall that $\lambda^\vee x\Omega_s$ is an abbreviation for $\langle \lambda^\vee x\Omega_s, \varepsilon\rangle$ and that $H \vdash \varepsilon : H$ by S1). Then $\vdash \mathbf{M}^s_\gamma : \gamma$ by L3 since $\gamma \to \omega \leqslant \gamma$ by $E_s5.1$. It is easy to see, using L_s5 twice, that $\vdash \mathbf{T}^s_\gamma : \gamma \to (\psi \to \psi)$.

$$\mathbf{M}_{\phi\wedge\psi} = (\mathbf{M}_\phi \parallel \mathbf{M}_\psi)$$
$$\mathbf{T}_{\phi\wedge\psi} = \lambda x(\mathbf{T}_\phi x)(\mathbf{T}_\psi x)$$

By the induction hypothesis we have $\vdash \mathbf{M}_\phi : \phi$ and $\vdash \mathbf{M}_\psi : \psi$, therefore by L7

$$\vdash (\mathbf{M}_\phi \parallel \mathbf{M}_\psi) : \phi \wedge \psi$$

as required. By the induction hypothesis, for any τ we have $\vdash \mathbf{T}_\phi : \phi \to (\tau \to \tau)$, hence by weakening,

$$x:\phi\wedge\psi \vdash \mathbf{T}_\phi : \phi \to (\tau \to \tau).$$

Moreover by L4 and L3, $x:\phi\wedge\psi \vdash x:\phi$. Then using L6,

$$x:\phi\wedge\psi \vdash \mathbf{T}_\phi x : \tau \to \tau.$$

Similarly we have $x:\phi\wedge\psi \vdash \mathbf{T}_\psi x : \xi \to \xi$. Let $\tau = (\xi \to \xi)$; then by L6,

$$x:\phi\wedge\psi \vdash (\mathbf{T}_\phi x)(\mathbf{T}_\psi x) : \xi \to \xi,$$

and finally $\vdash \lambda x(\mathbf{T}_\phi x)(\mathbf{T}_\psi x) : (\phi\wedge\psi) \to (\xi \to \xi)$ by L5.1, which is the required property of $\mathbf{T}_{\phi\wedge\psi}$. The proofs are entirely similar for

$$\mathbf{M}^s_{\phi\wedge\psi} = (\mathbf{M}^s_\phi \parallel \mathbf{M}^s_\psi)$$
$$\mathbf{T}^s_{\zeta\wedge\tau} = \lambda^\vee x(\mathbf{T}^s_\zeta x)(\mathbf{T}^s_\tau x).$$

The last case of the definition of the characteristic terms concerns implicative formulae:

$$\mathbf{M}_{\phi\to\psi} = \lambda x(\mathbf{T}_\phi x)\,\mathbf{M}_\psi$$
$$\mathbf{T}_{\phi\to\psi} = \lambda^\vee x(\mathbf{T}_\psi(x\mathbf{M}_\phi)).$$

By the induction hypothesis $x:\phi \vdash \mathbf{M}_\psi : \psi$. As we saw above

$$x:\phi \vdash \mathbf{T}_\phi x : \psi \to \psi.$$

Then by L6,

$$x:\phi \vdash (\mathbf{T}_\phi x)\,\mathbf{M}_\psi : \psi,$$

and by L5.1 we get the required property of $\mathbf{M}_{\phi\to\psi}$, namely,

$$\vdash \lambda x(\mathbf{T}_\phi x)\,\mathbf{M}_\psi : \phi\to\psi.$$

By the induction hypothesis $\vdash \mathbf{M}_\phi : \phi$, therefore by L4 and L6 (and weakening),

$$x:\phi\to\psi \vdash (x\mathbf{M}_\phi) : \psi.$$

Using the induction hypothesis on $\mathbf{T}_\psi$ we have, by weakening and L6,

$$x:\phi\to\psi \vdash \mathbf{T}_\psi(x\mathbf{M}_\phi) : \xi\to\xi.$$

Since $\phi\to\psi \leqslant_\star \omega\to\omega$ by $E_\star 7$ and $E_\star 5$, we get

$$\vdash \lambda^\vee x(\mathbf{T}_\psi(x\mathbf{M}_\phi)) : (\phi\to\psi)\to(\xi\to\xi)$$

by L5.2. For the strict calculus, we define

$$\mathbf{M}^s_{\zeta\to\phi} = \lambda^\vee x(\mathbf{T}^s_\zeta x)\,\mathbf{M}^s_\phi$$

$$\mathbf{T}^s_{\zeta\to\phi} = \begin{cases} \mathbf{T}^s_\gamma & \text{if } \phi\in\Phi_s - \Gamma \\ \lambda^\vee x\mathbf{T}^s_\phi(x\mathbf{M}^s_\zeta) & \text{otherwise.}\end{cases}$$

The proof that $\vdash \mathbf{M}^s_{\zeta\to\phi} : \zeta\to\phi$ proceeds exactly as for $\mathbf{M}_{\phi\to\psi}$. If $\phi\in\Phi_s-\Gamma$ then $\phi\sim\omega$, therefore $\zeta\to\phi\sim\gamma$. We saw that $\vdash \mathbf{T}^s_\gamma : \gamma\to(\psi\to\psi)$; hence in this case $\vdash \mathbf{T}^s_{\zeta\to\phi} : \gamma\to(\psi\to\psi)$ by L3. Otherwise, that is, if $\phi\in\Gamma$, we proceed as for $\mathbf{T}_{\phi\to\psi}$.

PROPOSITION 5.21.

(i) $H \Vdash \mathbf{M}_\phi : \psi \;\Rightarrow\; \phi\leqslant_\star\psi$ *and* $H\Vdash_s \mathbf{M}^s_\phi : \psi \Rightarrow \phi\leqslant_s\psi$.

(ii) $H\Vdash_\star \mathbf{T}_\phi : \psi\to(\zeta\to\tau) \;\Rightarrow\; \phi\geqslant_\star\psi$ *and* $\zeta\leqslant_\star\tau$ *and*

$H\Vdash_s \mathbf{T}^s_\phi : \psi\to(\zeta\to\tau) \;\Rightarrow\; \phi\geqslant_s\psi$ *and* $\zeta\leqslant_s\tau$.

Proof. By induction on the formula ϕ. Clearly we may assume that the sequents $H\Vdash_\star \mathbf{M}_\phi : \psi$ and $H\Vdash_s \mathbf{M}^s_\phi : \psi$ are not proved by means of T1, since in this case $\psi=\omega$, and obviously $\phi\leqslant\omega$ by E1.

• Assume that $H\Vdash_\star \mathbf{M}_\omega : \psi$, that is, $H\Vdash_\star \Omega : \psi$, with $\psi\not\sim_\star\omega$. Then we would have, by Lemma 5.18, $\psi\leqslant_\star\gamma$, hence $H\vdash_\star\Omega:\gamma$, by Proposition 5.20 and L3. But then by soundness we would have $\models_\star \Omega:\gamma$, and this is impossible since $\Omega\Uparrow$. Therefore $H\Vdash_\star \mathbf{M}_\omega : \psi\Rightarrow\omega\leqslant_\star\psi$. The proof is the same for $\mathbf{M}^s_\omega$.

• The sequent $H \Vdash_\star \mathbf{T}_\omega : \psi \to (\zeta \to \tau)$ can only be proved by means of T3.1; that is,

$$x : \psi,\ G \Vdash_\star \lambda yy : \zeta \to \tau$$

and $H \leqslant_\star G$ (recall that $H \vdash \varepsilon : G$ can only be inferred using S1). To prove this sequent one must use T3.1, thus $y : \zeta,\ F \Vdash_\star y : \tau$ with $(x : \psi, F) \leqslant_\star F$. Then by Remark 5.19 we have $\zeta \leqslant_\star \tau$, and obviously $\omega \geqslant_\star \psi$. The argument is the same for $\mathbf{T}^s_\gamma$. Note that if $H \Vdash_s \mathbf{T}^s_\gamma : \psi \to (\zeta \to \tau)$ we have $\psi \in \Gamma$, therefore $\gamma \geqslant_\star \psi$.

• The sequent $H \Vdash_s \mathbf{M}^s_\gamma : \psi$ can only be proved using L_s5 (or T_1); that is, $\psi = \zeta \to \tau$ with $x : \zeta,\ G \Vdash_s \Omega_s : \tau$. We have seen that this implies $\tau \sim_s \omega$, hence $\psi \sim_s \gamma$ by Lemma 5.18.

• The sequent $H \Vdash_\star (\mathbf{M}_\phi \parallel \mathbf{M}_\psi) : \tau$ is proved using T5.1 or T5.2 (or T1). In this last case, for instance, $\psi \leqslant_\star \tau$ by the induction hypothesis; therefore $\phi \wedge \psi \leqslant_\star \tau$ by E3.3 and E2. For T5.1 one uses E3.2. The proof is the same for $H \Vdash_s (\mathbf{M}^s_\phi \parallel \mathbf{M}^s_\psi) : \tau$.

• The sequent $H \Vdash_\star \mathbf{T}_{\phi \wedge \psi} : \delta \to (\zeta \to \tau)$ is proved by means of T3.1; thus

$$x : \delta,\ G \Vdash_\star (\mathbf{T}_\phi x)(\mathbf{T}_\psi x) : \zeta \to \tau$$

with $H \leqslant_\star G$. Since this must be proved using T4, there exist ν and $\nu_1, \ldots, \nu_m$ such that $\nu_1 \wedge \cdots \wedge \nu_m \leqslant_\star \nu$ and

$$x : \delta,\ G \Vdash_\star \mathbf{T}_\psi x : \nu_i \tag{1i}$$

$$x : \delta,\ G \Vdash_\star \mathbf{T}_\phi x : \nu \to (\zeta \to \tau). \tag{2}$$

This second sequent is proved by T4, thus:

$$x : \delta,\ G \Vdash_\star x : \mu_j \qquad \text{with} \quad \mu_1 \wedge \cdots \wedge \mu_r \leqslant_\star \mu \tag{3}$$

$$x : \delta,\ G \Vdash_\star \mathbf{T}_\phi : \mu \to (\nu \to (\zeta \to \tau)). \tag{4}$$

Then $\delta \leqslant_\star \mu$ by Remark 5.19. Moreover $\mu \leqslant_\star \phi$ by the induction hypothesis regarding (4), hence $\delta \leqslant_\star \phi$ (recall that we have to show $\phi \wedge \psi \geqslant_\star \delta$ and $\zeta \leqslant_\star \tau$), and $\nu \leqslant_\star \zeta \to \tau$ (induction hypothesis), hence $\nu_1 \wedge \cdots \wedge \nu_m \leqslant_\star \zeta \to \tau$. Then since each ν_i is irreducible, as well as τ, from Lemma 5.17 $\nu_i \ll_\star \zeta \to \tau$ for some i. It is easily seen that this implies that $\nu_i = \xi \to \pi$ with $\xi \geqslant_\star \zeta$ and $\pi \ll_\star \tau$. Therefore the sequent (1i) is proved using T4; that is,

$$x : \delta,\ G \Vdash_\star x : \eta_h \qquad \text{with} \quad \eta_n \wedge \cdots \wedge \eta_r \leqslant \eta$$

$$x : \delta,\ G \Vdash_\star \mathbf{T}_\psi : \eta \to (\xi \to \pi).$$

By the induction hypothesis $\eta \leqslant_\star \psi$, and by Remark 5.19, $\delta \leqslant_\star \eta_h$, hence

$$\delta \leqslant_\star \eta_1 \wedge \cdots \wedge \eta_r \leqslant_\star \eta \leqslant_\star \psi,$$

and finally $\delta \leqslant_\star \phi \wedge \psi$. Moreover, by the induction hypothesis $\xi \leqslant_\star \pi$, therefore $\xi \leqslant_\star \xi \leqslant_\star \pi \leqslant_\star \tau$; that is, $\zeta \leqslant_\star \tau$. The proof is the same in the strict case, i.e., for $H \Vdash_s \mathbf{T}^s_{\phi \wedge \psi} : \delta \to (\zeta \to \tau)$.

• If $H \Vdash_\star \mathbf{M}_{\phi \to \psi} : \zeta$, then this sequent is proved by means of T3 (or T1); that is, $\zeta = \xi \to \tau$ with

$$x : \xi,\ G \Vdash_\star (\mathbf{T}_\phi x)\, \mathbf{M}_\psi : \tau,$$

where $H \leqslant_\star G$. If this sequent is proved using T1 then $\tau = \omega$ and we have $\phi \to \psi \leqslant_\star \zeta$ since

$$\phi \to \psi \leqslant_\star \phi \to \omega \leqslant_\star \omega \to \omega \leqslant_\star \xi \to \omega.$$

Otherwise there exist ν and $\nu_1, \ldots, \nu$ such that $\nu_1 \wedge \cdots \wedge \nu_m \leqslant \nu$ and

$$x : \xi,\ G \Vdash_\star \mathbf{M}_\psi : \nu_i$$
$$x : \xi,\ G \Vdash_\star \mathbf{T}_\phi x : \nu \to \tau.$$

By the induction hypothesis $\psi \leqslant_\star \nu_i$ for all i, therefore $\psi \leqslant_\star \nu$. The second sequent is proved by T4; thus:

$$x : \xi,\ G \Vdash_\star x : \mu_j \qquad \text{with} \quad \mu_1 \wedge \cdots \wedge \mu_n \leqslant \mu$$
$$x : \xi,\ G \Vdash_\star \mathbf{T}_\phi : \mu \to (\nu \to \tau).$$

Then $\xi \leqslant_\star \mu_j$ by Remark 5.19; hence $\xi \leqslant_\star \mu$. Moreover $\mu \leqslant_\star \phi$ and $\nu \leqslant_\star \tau$ by the induction hypothesis, hence $\xi \leqslant_\star \phi$ and $\psi \leqslant_\star \tau$, which implies that $\phi \to \psi \leqslant_\star \xi \to \tau = \xi$. The proof is entirely similar in the strict case, that is, for $H \Vdash_s \mathbf{M}^s_{\phi \to \psi} : \zeta$. Note that if $\zeta = \xi \to \tau$ with $\tau = \omega$ then $\phi \to \psi \leqslant_s \zeta$ by $E_s5.2$ and E_s7 (and $E_s5.1$).

• The sequent $H \Vdash_\star \mathbf{T}_{\phi \to \psi} : \theta \to (\zeta \to \tau)$ can only be inferred by means of T3.2; that is,

$$x : \theta,\ G \vdash_\star \mathbf{T}_\psi (x \mathbf{M}_\phi) : \zeta \to \tau,$$

with $H \leqslant_\star G$ and $\theta \leqslant_\star \gamma$. Then this sequent is proved using T4, and there exist ν and $\nu_1, \ldots, \nu_k$ such that $\nu_1 \wedge \cdots \wedge \nu_k \leqslant_\star \nu$ and

$$x : \theta,\ G \Vdash_\star (x \mathbf{M}_\phi) : \nu_i \tag{1i}$$
$$x : \theta,\ G \Vdash_\star \mathbf{T}_\psi : \nu \to (\zeta \to \tau). \tag{2}$$

By the induction hypothesis on (2) we have $\psi \geqslant_\star \nu$ and $\zeta \leqslant_\star \tau$. It remains to show that $\phi \to \psi \geqslant_\star \theta$. We distinguish two cases:

• • If $\psi \sim \omega$ then $\phi \to \psi \sim \gamma$ (Lemma 5.18), and we saw that $\theta \leqslant_\star \gamma$ (this was the side condition associated with the use of T3.2 in proving $H \Vdash_\star \mathbf{T}_{\phi\to\psi} : \theta \to (\zeta \to \tau)$).

• • Otherwise we have $\nu \not\sim_\star \omega$, therefore the set $\{\delta_1, ..., \delta_n\} = \{\nu_1, ..., \nu_k\} - \{\omega\}$ is non-empty, and

$$\nu_1 \wedge \cdots \wedge \nu_k \sim_\star \delta_1 \wedge \cdots \wedge \delta_n \leqslant_\star \nu.$$

The sequent

$$x : \theta,\ G \Vdash_\star (x\mathbf{M}_\phi) : \delta_j$$

can only be inferred by means of T4; therefore we have

$$x : \theta,\ G \Vdash_\star x : \eta_j \to \delta_j$$
$$x : \theta,\ G \Vdash_\star \mathbf{M}_\phi : \eta_i^j \qquad \text{with} \quad \eta_1^j \wedge \cdots \wedge \eta_{m_j}^j \leqslant_\star \eta_j.$$

By the induction hypothesis $\phi \leqslant_\star \eta_i^j$, hence $\phi \leqslant_\star \eta_j$. This implies that $\eta_j \to \delta_j \leqslant_\star \phi \to \delta_j$, therefore

$$\bigwedge_{1\leqslant j\leqslant n} (\eta_j \to \delta_j) \leqslant_\star \bigwedge_{1\leqslant j\leqslant n} (\phi \to \delta_j) \sim_\star \Big(\phi \to \bigwedge_{1\leqslant j\leqslant n} \delta_j\Big)$$
$$\leqslant_\star \phi \to \nu \leqslant_\star \phi \to \psi.$$

By Remark 5.19 we have $\theta \leqslant_\star \eta_j \to \delta_j$ for any j, therefore $\theta \leqslant_\star \phi \to \psi$.

The proof is the same for the strict case; that is, $H \vdash_s \mathbf{T}^s_{\phi\to\psi} : \theta \to (\zeta \to \tau)$. Note that if $\psi \sim_s \omega$ then $\psi \in \Phi_s - \Gamma$ by Lemma 5.18; in this case $\mathbf{T}^s_{\phi\to\psi} = \mathbf{T}^s_\gamma$, and this was already treated. ▮

Finally we can establish the characterization lemma:

COROLLARY 5.22 (Characterization Lemma). *For any formula* ϕ

$$\vdash_\star \mathbf{M}_\phi : \psi \Leftrightarrow \phi \leqslant_\star \psi \qquad \textit{and} \qquad \vdash_s \mathbf{M}^s_\phi : \psi \Leftrightarrow \phi \leqslant_s \psi.$$

Proof. We have seen that $\vdash_\star \mathbf{M}_\phi : \phi$; therefore if $\phi \leqslant_\star \psi$ we have $\vdash_\star \mathbf{M}_\phi : \psi$ by L3. Conversely if $\vdash_\star \mathbf{M}_\phi : \psi$ then by Proposition 5.20 there exist $\psi_1, ..., \psi_k$ such that

$$\psi_1 \wedge \cdots \wedge \psi_k \leqslant \psi \qquad \text{and} \qquad \forall i.\ \vdash_\star \mathbf{M}_\phi : \psi_i.$$

Then we have $\phi \leqslant_\star \psi_i$ for all i be the previous proposition, hence $\phi \leqslant_\star \psi$. The proof is the same in strict case. ▮

ACKNOWLEDGMENTS

Merci à Jean-Jacques Lévy: l'enseignement tiré de discussions tenues avec lui m'a été fort utile pour l'élaboration de ce travail. Thanks also to the referees for their helpful comments.

RECEIVED December 7, 1990; FINAL MANUSCRIPT RECEIVED October 16, 1991

REFERENCES

1. ABADI, M., CARDELLI, L., CURIEN, P.-L., AND LÉVY, J.-J. (1990), Explicit substitutions, *in* "POPL 90," pp. 31–46.
2. ABRAMSKY, S. (1987), Domain theory in logical form, *in* "LICS 87," 47–53.
3. ABRAMSKY, S. (1989), The lazy lambda-calculus, *in* "Research Topics in Functional Programming" (D. Turner, Ed.), pp. 65–116, Addison–Wesley, Reading, MA.
4. ABRAMSKY, S., AND ONG, C.-H. L. (1989/1993), "Full Abstraction in the Lazy Lambda-calculus," Research Report, Dept. of Computing, Imperial College; *Inform. and Comput.* **105**, 159–267.
5. BARENDREGT, H. (1984), "The Lambda Calculus," Studies in Logic, Vol. 103, Revised ed., North Holland, Amsterdam.
6. BARENDREGT, H., COPPO, M., AND DEZANI-CIANCAGLINI, M. (1983), A filter lambda model and the completeness of type assignment, *J. Symbolic Logic* **48**, 931–940.
7. BERRY, G., CURIEN, P.-L., AND LÉVY, J.-J. (1985), Full abstraction for sequential languages: The state of the art, *in* "Algebraic Methods in Semantics" (M. Nivat and J. C. Reynolds, Eds.), pp. 90–132, Cambridge Univ. Press, London/New York.
8. BLOOM, B., AND RIECKE, J. G. (1989), LCF should be lifted, *in* "Conference on Algebraic Methodology and Software Technology, University of Iowa," pp. 133–136.
9. BLOOM, B. (1990), Can LCF be topped? Flat lattice models of typed lambda-calculus, *Inform. and Comput.* **87**, 264–301.
10. COPPO, M., DEZANI-CIANCAGLINI, M., AND VENNERI, B. (1981), Functional characters of solvable terms, *Z. Math. Logik Grundlag. Math.* **27**, 45–58.
11. COPPO, M., DEZANI-CIANCAGLINI, M., AND VENNERI, B. (1980), Principal type schemes and lambda-calculus semantics, *in* "To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism" (J. R. Hindley and J. P. Seldin, Eds.), pp. 535–560, Academic Press, New York.
12. COPPO, M., DEZANI-CIANCAGLINI, M., HONSELL, F., AND LONGO, G. (1984), Extended type structures and filter lambda models, *in* "Logic Colloquium 82," pp. 241–262, North-Holland, Amsterdam.
13. COPPO, M., DEZANI-CIANCAGLINI, M., AND LONGO, G. (1983), Applicative information systems, *in* "CAAP83," Lecture Notes in Computer Science, Vol. 159, pp. 35–64, Springer-Verlag, Berlin/New York.
14. COPPO, M. (1984), Completeness of type assignment in continuous lambda models, *Theoret. Comput. Sci.* **29**, 309–324.
15. COSMADAKIS, S., MEYER, A. R., AND RIECKE, J. G. (1990), Completeness for typed lazy inequalities, *in* "LICS 90," pp. 312–320.
16. CURIEN, P.-L. (1986), "Categorical Combinators, Sequential Algorithms and Functional Programming," Research Notes in Theoretical Computer Science, Pitman, London/Boston.
17. DEZANI-CIANCAGLINI, M., AND MARGARIA, I. (1986), A characterization of *F*-complete type assignments, *Theoret. Comput. Sci.* **45**, 121–157.
18. SITARAM, D., AND FELLEISEN, M. (1990), Reasoning with continuations. II. Full abstraction for models of control, *in* "1990 ACM Conference on Lisp and Functional Programming," pp. 161–175.

19. HINDLEY, R. (1983), The completeness theorem for typing λ-terms, *Theor. Comput. Sci.* **22**, 1–17, 127–133.
20. HINDLEY, R. (1982), The simple semantics for Coppo–Dezani–Sallé types, *in* "International Symposium on Programming," Lecture Notes in Computer, Science, Vol. 137, pp. 212–226, Springer-Verlag, Berlin/New York.
21. KRIVINE, J.-L. (1990), "Lambda-Calcul: Types et Modèles," Masson, Paris.
22. LARSEN, K. G., AND WINSKEL, G. (1984), Using information systems to solve recursive domain equations effectively, *in* "Semantics of Data Types," Lecture Notes in Computer Science, Vol. 173, pp. 109–130, Springer-Verlag, Berlin/New York.
23. LEIVANT, D. (1986), Typing and computational properties of lambda expressions, *Theoret. Comput. Sci.* **44**, 51–68.
24. LÉVY, J.-J. (1976), An algebraic interpretation of the $\lambda\beta K$-calculus; and an application of a labelled λ-calculus, *Theoret. Comput. Sci.* **2**, 97–114.
25. LONGO, G. (1983), Set-theoretical models of λ-calculus: Theories, expansions, isomorphisms, *Ann. Pure Appl. Logic* **24**, 153–188.
26. MEYER, A. R. (1988), Semantical paradigms, *in* "LICS 88," pp. 236–253.
27. MILNER, R. (1973), Processes: A mathematical model of computing agents, *in* "Logic Colloquium 73," pp. 157–173, North-Holland, Amsterdam.
28. MILNER, R. (1977), Fully abstract models of typed λ-calculi, *Theoret. Comput. Sci.* **4**, 1–22.
29. MOGGI, E. (1986), Categories of partial morphisms and the λ_p-calculus, *in* "Lectures Notes in Computer Sciences," Vol. 240, pp. 242–251, Springer-Verlag, Berlin/New York.
30. MOGGI, E. (1989), Computational λ-calculus and monads, *in* "LICS 89," pp. 14–23.
31. NIELSEN, M., PLOTKIN, G., AND WINSKEL, G. (1981), Petri nets, event structures and domains, *Theoret. Comput. Sci.* **13**, 85–108.
32. ONG C.-H. L. (1988), "The Lazy Lambda-Calculus: An Investigation into the Foundations of Functional Programming," Ph.D. Thesis, Department of Computing, Imperial College.
33. ONG, C.-H. L. (1988), Fully abstract models of the lazy lambda calculus, *in* "29th FOCS, pp. 368–376.
34. PLOTKIN, G. (1975), Call-by-name, call-by-value and the λ-calculus, *Theoret. Comput. Sci.* **1**, 125–159.
35. PLOTKIN, G. (1977), LCF considered as a programming language, *Theoret. Comput. Sci.* **5**, 223–256.
36. PLOTKIN, G. (1978), T^{ω} as a universal domain, *J. Comput. System Sci.* **17**, 209–236.
37. PLOTKIN, G. (1985), "Types and Partial Functions. A Metalanguage. (Towards a) Logic for Computable Functions," CSLI Summer School Notes, manuscript.
38. RIECKE, J. G. (1990), A complete and decidable proof system for call-by-value equalities, *in* "ICALP 90," Lecture Notes in Computer Science, Vol. 443, pp. 20–31.
39. SCOTT, D. (1970), Outline of a mathematical theory of computation, *in* "Fourth Annual Princeton Conference on Information Sciences and Systems," pp. 169–176.
40. SCOTT, D. (1976), Data types as lattices, *SIAM J. Comput.* **5**, 522–587.
41. SCOTT, D. (1979), Identity and existence in intuitionistic logic, *in* "Application of Sheaves," Lecture Notes in Mathematics, Vol. 753, pp. 660–696.
42. SCOTT, D. (1980), Lambda-calculus: Some models, some philosophy, *in* "The Kleene Symposium," pp. 223–265, North-Holland, Amsterdam.
43. SCOTT, D. (1982), Domains for denotational semantics, *in* "ICALP 82," Lecture Notes in Computer Science, Vol. 140, pp. 577–613, Springer-Verlag, Berlin/New York.
44. STOUGHTON, A. (1988), "Fully Abstract Models of Programming Languages," Research Notes in Theoretical Computer Science, Pitman, London/Boston.
45. WADSWORTH, C. (1976), The relation between computational and denotational properties for Scott D_∞-models of the lambda-calculus, *SIAM J. Comput.* **5**, 488–521.